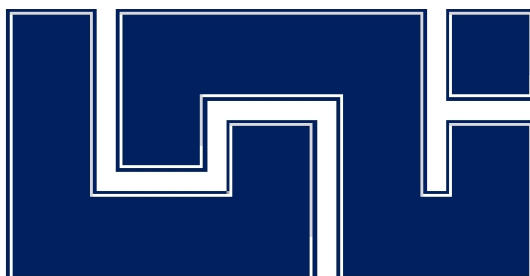


Mon
006.3
G643
2010

UNIVERSIDAD NACIONAL DE INGENIERÍA
Recinto Universitario Simón Bolívar
Facultad de Electrotecnia y Computación



Líder en Ciencia y Tecnología

**Implementación y evaluación de los algoritmos de
Aprendizaje Automatizado propuestos en la asignatura
Inteligencia Artificial**

TRABAJO MONOGRÁFICO

PRESENTADO POR:

Adilson Isaac González López
Danny Oswaldo Chávez Miranda

TUTOR:

Lic. Virginia Herrera Pinell

PARA OPTAR AL TÍTULO DE:
Ingeniero en Computación

Managua, Nicaragua
Octubre, 2010

Dedicatoria

A mí apreciada madre Vicentina López
Gracias mami!!!
Adilson G. López

Se lo dedico a toda mi familia que me han
apoyado de manera incondicional
Danny Chávez

Agradecimientos

*A*gradecemos A:

- ✓ *N*uestra Tutora Virginia Herrera Pinel, catedrático de la Universidad Nacional de Ingeniería. Por guiarnos y compartir sus conocimientos y vasta experiencia, elementos imprescindibles para la culminación de nuestro trabajo.
- ✓ *A*l Licenciado Elmer Ramírez España Coordinador del Programa de Iniciación al Ejercicio Docente (PIED), que en conjunto con las autoridades de la Facultad de Electrotecnia y Computación FEC, nos brindaron recursos e insumos económicos para llevar a cabo nuestro estudio monográfico.
- ✓ *A*nuestros compañeros de universidad y aquellas personas que fueron parte de éste proceso de desarrollo importante de nuestras vidas
- ✓ *A*aquellos profesores que durante la carrera compartieron sus experiencias con el fin de lograr un buen proceso de enseñanza – aprendizaje, y aquellos docentes que aparte de su contribución intelectual nos ayudaron a crecer como personas.

RESUMEN

El presente trabajo monográfico consiste en la implementación y evaluación de los algoritmos de aprendizaje automatizado, también llamados aprendizaje de máquina. Los algoritmos que presentamos son actualmente impartidos en la asignatura de inteligencia artificial en el IX semestre de la carrera de Ingeniería en Computación.

La implementación de los algoritmos de aprendizaje automatizado que se abordarán en éste estudio monográfico permitirá al lector comprender el mecanismo de trabajo de cada uno de los algoritmos en cuestión, además se tendrá un conjunto de datos para que podamos ver la funcionalidad de los mismos.

Los algoritmos de aprendizaje Automatizado tienen como objetivo principal generalizar comportamientos a partir de información suministrada y no estructurada en forma de casos o ejemplos. Los algoritmos considerados son 3 que corresponden a:

- 1) Técnica de Naive Bayes.
- 2) Árboles de Decisión.
- 3) El Vecino más cercano.

Además se utilizó la técnica de validación cruzada para el análisis de información.

El desarrollo de este tema pretende que sea de utilidad para los estudiantes de la carrera de Ingeniería en Computación como fuente de referencia bibliográfica para la asignatura de Inteligencia artificial, además pretende favorecer a las personas interesadas en este campo de estudio, mediante la lectura y seguimiento de la evaluación y resultados de los casos de estudios mostrados en este escrito.

Para la implementación y evaluación de los algoritmos, se consideró la base de datos de registros de notas de los estudiantes de la carrera de ingeniería en computación, considerando aquí los registros comprendidos en el periodo del I Semestre de año lectivo 2006 hasta el II semestre del año lectivo 2009. La base de datos fue proporcionada por la División de informática y tecnología de la información (DITI-UNI).

La implementación fue realizada en Microsoft Visual Studio 2005, y como gestor de base de datos SQL Server Managment Studio 2005

Índices

Índice General

1. Aspectos Metodológicos.....	1
1.1 Introducción.....	1
1.3 Objetivos.....	3
1.3.1 Objetivo General:.....	3
1.3.2 Objetivos Específicos:.....	3
1.2 Justificación.....	4
1.4 Diseño Metodológico.....	5
2. Marco Teórico.....	7
2.1 Aprendizaje Automatizado.....	7
2.2 Aprendizaje Supervisado.....	8
2.3 Aprendizaje por Agrupamiento (Clustering).....	8
2.4 Aprendizaje por Refuerzo (Reinforcement).....	8
2.5 Aspectos de la función de Aprendizaje.....	8
2.5.1 Memorizando.....	9
2.5.2 Promediando.....	9
2.5.3 Generalización.....	11
2.6 Aprendizaje Supervisado.....	15
2.6.1 La mejor hipótesis.....	16
2.6.2 Aprendizaje como búsqueda.....	17
2.6.3 Validación cruzada.....	18
3. Métodos de Aprendizaje.....	22
3.2 Vecino más cercano (Lazy Learning).....	22
3.3 Árboles de Decisión.....	25
3.3.1 Árboles de decisión y aprendizaje Automatizado.....	27
3.3.2 Algoritmo para Construcción de árboles decisión.....	29
3.3.3 Escogiendo el mejor atributo.....	30
3.3.4 Simulación del Algoritmo.....	34
3.4 Naive Bayes.....	37

3.4.1 Algoritmo.....	40
3.4.2 Naive Bayes y Corrección de Laplace.....	41
4. Implementación, Evaluación y Resultados.....	44
4. 1 Datos.....	45
4.1.2 Suministración de nuestro conjunto de datos.....	45
4.2 Técnica de aprendizaje: Naive Bayes.....	47
4.2.1 Propósito al implementar esta técnica.....	47
4.2.2 Ventajas de la Técnica.....	47
4.2.3 Desventajas de la Técnica.....	47
4.2.4 Implementación de la Técnica.....	47
4.2.4.1 Caso de Estudio.....	48
4.2.4.2 Datos de Entrenamiento.....	48
4.2.4.3 Consideraciones Generales del Conjunto de Datos.....	49
4.2.4.4 Seleccionando Nuestro Conjunto de Entrenamiento con IMEAN 1.0.....	49
4.2.4.5 Transformación de Datos con IMEAN 1.0.....	50
4.3 Algoritmo de Aprendizaje: Árboles de Decisión.....	58
4.3.1 Propósito al implementar esta técnica.....	58
4.3.3 Desventajas.....	58
4.3.4 Implementación del Algoritmo.....	58
4.3.4.1 Caso Estudio.....	58
4.3.4.2 Consideraciones Generales del Conjunto de Datos.....	60
4.3.4.3 Seleccionando Nuestros Conjunto de Entrenamiento IMEAN 1.0.....	60
4.3.4.4 Transformación de Datos con IMEAN 1.0.....	61
4.4 Técnica de Aprendizaje: Vecino más Cercano.....	65
4.4.1 Propósito al implementar esta técnica.....	65
4.4.2 Ventajas de la Técnica:.....	65
4.4.3 Desventajas:.....	65
4.4.4 Implementación del Algoritmo.....	65
4.4.4.1 Caso Estudio.....	65
4.4.4.2 Consideraciones Generales del Conjunto de Datos.....	66

4.4.4.3 Seleccionando Nuestros Conjunto de Entrenamiento con IMEAM 1.0.	67
4.4.4.4 Transformación de Datos con IMEAM 1.0	68
4.4.9 Vecino Cercano y Distancia Manhattan.	73
4.5 Validación Cruzada.	79
4.5.1 Propósito.....	79
4.5.3 Caso de Estudio.	80
CONCLUSIONES.....	84
RECOMENDACIONES.	85
BIBLIOGRAFÍA	86
ANEXOS.....	88
Anexo A: Manual de usuario e interfaces gráficas.....	88
Anexo B: Código fuente de implementación.....	107

Aspectos Metodológicos

Cap. 1

1. Aspectos Metodológicos.

1.1 Introducción

Inteligencia Artificial es un campo muy relacionado con muchas áreas y disciplinas, tales como la teoría de probabilidad y estadística, minería de datos, reconocimiento de patrones, control adaptativo, neurociencia computacional e informática teórica.

El aprendizaje automatizado es una rama de la inteligencia artificial, ésta última se centra en crear algoritmos, procesos, programas que sean capaces de mostrar un comportamiento que sea considerado como “Inteligente”.

El Aprendizaje de máquina o aprendizaje automatizado es una disciplina científica que tiene que ver con el diseño y desarrollo de algoritmos que permiten encontrar patrones de comportamientos en datos empíricos, a partir de datos de sensores o desde bases de datos. Se puede tomar ejemplos (casos) para capturar las características de interés en cuanto a la información almacenada.

Los datos pueden ser vistos como ejemplos que ilustran las relaciones entre las variables observadas (o atributos observados). Uno de los objetivos principales del aprendizaje automatizado en relación a la investigación es aprender a reconocer automáticamente patrones de comportamientos y tomar decisiones inteligentes sobre la base de datos, la dificultad radica en el hecho de que el conjunto de casos puede ser demasiado grande por el conjunto de entrenamiento. Por lo tanto se debe generalizar a partir de los ejemplos dados, con el fin de ser capaz de producir una salida útil de los nuevos casos.

Con este trabajo monográfico se pretende aplicar los distintos aprendizajes obtenidos durante la carrera de Ingeniería en Computación en relación a ciertas áreas de estudio, tales como algoritmos y estructura de datos, estadísticas, ingeniería de software e inteligencia artificial ya que se exponen en este estudio algunos fundamentos relacionado con el estudio en mención.

La documentación monográfica está estructurada en 5 capítulos y una parte de anexos.

El capítulo1 comprende los elementos esenciales que dieron origen al desarrollo de éste trabajo como son: Objetivos, justificación del tema y diseño metodológico.

El capítulo 2 es una introducción general al Aprendizaje Automatizado como un campo de la inteligencia artificial, también se mencionan los orígenes, el interés de estudio del campo y la importancia dentro del campo informático y su impacto social. Además se abordarán los conceptos fundamentales que caracterizan el aprendizaje Automatizado.

En el capítulo 3 abordaremos las técnicas de aprendizaje de máquina en la que se especifica mecanismos de trabajo, importancia, así como la implicancia de dichas técnica con otras disciplinas, las técnicas son “El vecino más cercano”, “Arboles de decisión”, y finalmente conoceremos “Naive Bayes” como otra técnica de aprendizaje. Una vez abordadas las técnicas de aprendizaje se mostrará capítulo 4 que contiene las evaluaciones y resultado de cada algoritmo

Finalmente una parte de anexos, fuentes bibliográficas y conclusiones del estudio.

1.3 Objetivos.

1.3.1 Objetivo General:

Desarrollar una **aplicación de software** que implemente las técnicas de Aprendizaje Automatizado (Vecino más cercano, Naive Bayes y Árboles de Decisión), **ejecutarlas** con las notas estudiantiles y **aplicar** la técnica de Validación Cruzada.

1.3.2 Objetivo Específico:

1. Analizar y Seleccionar las notas del caso de estudio en los que se aplicarán cada algoritmo.
2. Diseñar un sistema con las técnicas Vecino más cercano, Naive Bayes y con el algoritmo de Árboles de Decisión.
3. Implementar la función que transforma las notas, pre procesamiento de datos.
4. Implementar y aplicar cada algoritmo de aprendizaje automatizado sobre cada caso de estudio definido.
5. Validar la exactitud de la hipótesis obtenida con la Técnica de Validación Cruzada para el algoritmo de Naive Bayes.
6. Analizar y documentar resultados del caso de estudio mostrando la factibilidad y confiabilidad de las técnicas de Aprendizaje Automatizado.

1.2 Justificación.

Productos comerciales como Data Warehouse, Data Mining y recientemente Inteligencia Empresarial o Inteligencia de Negocios han interesado a las empresas, porque sondan y exploran datos para sacar información oculta. Esas metodologías incorporan técnicas del área Aprendizaje de máquina, abordadas en la asignatura de Inteligencia Artificial, tales como la técnica de Naive Bayes, la técnica de vecino más cercano y el algoritmo de árboles de decisión.

Dominar estas técnicas para implementar software o utilizar software especializado con estas características, abre un campo de trabajo profesional para ingenieros en computación.

Por consiguiente este estudio monográfico es una implementación y evaluación de los algoritmos de aprendizaje automatizado, con la finalidad de explicar paso a paso estas técnicas básicas de aprendizaje automatizado que a diferencia de algunas propuestas de software libre carecen de estas explicaciones

Plantear y documentar el resultado de esta investigación facilitará la comprensión de las ventajas y oportunidades que ofrecen estas metodologías de aprendizaje del campo de la inteligencia artificial para comprender software avanzados como los mencionados anteriormente y poder aplicarlos en las empresas.

Sin duda, esperamos que este trabajo contribuya a aumentar de forma significativa el interés de los estudiantes con respecto a la asignatura de Inteligencia Artificial, y que la utilicen como un medio de referencia bibliográfica tanto para los docentes y estudiantes de la carrera de ingeniería en computación.

1.4 Diseño Metodológico.

El estudio a realizar es de forma experimental y de carácter investigativo, pues estudiaremos cada algoritmo de Aprendizaje Automatizado y su aplicación a datos reales (notas de alumnos de la Carrera de Ingeniería en Computación, registros del periodo desde 2006 hasta 2009).

Este proceso involucra:

- ✓ Análisis y selección de datos.
- ✓ Desarrollar un software que incorpore las técnicas de Aprendizaje automatizado (VMC, NB, AD), que permita encontrar la hipótesis que definirán distintos desempeños de los estudiantes de la carrera de ingeniería en computación, carrera ubicada en la Facultad de Electrotecnia y Computación (FEC) en relación de las notas.
- ✓ Transformar las notas de los estudiantes de la carrera de ingeniería en computación en archivos binarios para ser utilizados en las técnicas de Aprendizaje Automatizado.

Para el desarrollo del software en el que serán implementados los algoritmos de aprendizaje se utilizarán las plataformas Visual Basic Net, SQL server.

- ✓ Presentar gráfica que muestre la eficiencia y eficacia de los algoritmos con la técnica de validación cruzada.
- ✓ Analizar recursos de software utilizados Visual Studio y Microsoft SQL Server Studio.
- ✓ Los algoritmos de aprendizaje de maquinas permiten encontrar patrones de comportamiento ante un conjunto de datos para la predicción de nuevos casos.

Aprendizaje Automatizado

Cap. 2

2. Marco Teórico.

2.1 Aprendizaje Automatizado (Automatizado).

Construir sistemas que tengan conocimiento representado explícitamente dentro de ellos, es una tarea de Inteligencia Artificial y es abordado en el campo de Aprendizaje Automatizado. Una forma de adquirir ese conocimiento es incorporarlo manualmente. Pero esto puede estar propenso al consumo excesivo de tiempo y errores.

Pero, en la mayoría de los casos, cuando no tenemos acceso directo a una descripción formal de un problema, podemos aprender algo acerca de ello desde ejemplos.

La palabra aprendizaje tiene diferentes significados. Es usado, al menos, para describir:

1. Memorización de algo
2. Aprendizaje de hechos a través de la observación y la exploración
3. Desarrollo de habilidades motoras y/o cognitivas a través de la práctica
4. Organización de nuevo conocimiento mediante representaciones efectivas

Herb Simon (una figura histórica importante en I.A. y en Economía) da la siguiente definición de aprendizaje:

“Aprender denota cambios en los sistemas que son adaptativos en el sentido que esto le permiten al sistema hacer la tarea o tareas con características parecidas más eficientemente y más efectivamente la próxima vez”.

Esto no es completamente preciso, pero nos da la idea de que los sistemas de aprendizaje tienen que adquirir alguna información de ejemplos, de un problema, y desempeñarse mejor debido a eso (así un sistema que simplemente registra todas las imágenes alguna vez vistas no lo llamaríamos un sistema de aprendizaje, porque aunque, en cierto sentido, el sabe mucho, no puede aprovecharse de su conocimiento).

2.2 Aprendizaje Supervisado.

El **aprendizaje supervisado** es el problema de aprendizaje más común. Al tener los pesos y longitudes individuales de un grupo de peces tipo salmón y los pesos y longitudes de un grupo de peces tipo atún. El trabajo de un sistema de aprendizaje supervisado será encontrar la regla predictiva que, dado el peso y longitud de un pescado, prediga si es salmón o atún.

2.3 Aprendizaje por Agrupamiento (Clustering).

Otro, menos definido problema de aprendizaje es el **clustering**. Ahora se tiene la descripción de un grupo de animales diferentes (o estrellas, o documentos) en términos de un conjunto de características (peso, número de patas, presencia de pelos, etc.) y el trabajo es dividirlos en grupos o posiblemente en una jerarquía de grupos que “tengan sentido”. Lo que hace la diferencia con el aprendizaje supervisado es que no se dice por adelantado que grupos de animales se deben hacer, se tiene que encontrar un agrupamiento natural.

2.4 Aprendizaje por Refuerzo (Reinforcement).

Otro problema de aprendizaje, muy familiar, es el aprendizaje de la habilidad motora, como andar en bicicleta. Llamamos a esto **aprendizaje por refuerzo**. Es diferente a los anteriores porque no se tiene un conjunto de información o conjunto de entrenamiento y su mecanismo de aprendizaje es mediante los aciertos y fallas.

El aprendizaje supervisado será el centro de estudio de este material monográfico el cual nos permitirá conocer las ideas fundamentales de aprendizaje de máquina

Otra forma de abordar el tema del aprendizaje es como tratar de encontrar la definición de una función, dado un conjunto de ejemplos con sus entradas y salidas. Esto podría verse como una definición bien ajustada, pero esto actualmente cubre una gran cantidad de casos.

2.5 Aspectos de la función de Aprendizaje.

El problema de aprendizaje como una función desde los ejemplos, es complicado. Se puede pensar en al menos tres diferentes problemas involucrados:

1. Memorizar
2. Promediar
3. Generalizar.

2.5.1 Memorizando.

Imagine que intentamos predecir si mi vecina va a manejar para ir al trabajo mañana, de tal manera que me lleve a mí también. Si ella va a manejar parece depender de los siguientes atributos:

1. La Temperatura.
2. El Tipo de precipitación.
3. El día de la semana.
4. Si necesita comprar en el camino a casa.
5. El Tipo de Ropa.

Temperatura	Precipitación	Día	Compra	Ropa	Salida.
80	Ninguna	Sábado	No	Casual	Camina
19	Nieve	Lunes	Si	Casual	Maneja
65	Ninguna	Miércoles	No	Casual	Camina

Tabla2.1 Datos recopilados por observación.

Se observa a nuestra vecina por tres días, los cuales están descritos en la tabla 1.1, la cual especifica las propiedades de los días y si la vecina caminó o manejó.

Ahora, nos encontramos con un día nevado, 19 grados, Lunes, cuando la vecina esta vistiendo ropa casual y va de compras. ¿Manejará? Ver tabla1.2.

La respuesta estándar es “si”, ese día es igual a uno de los vistos antes, y por tanto parece ser una buena apuesta predecir que “Si”. Esta es la forma más rudimentaria de aprender, lo cual es solamente memorizar las cosas que se ha visto antes. Aun así, esto puede ayudar a hacer un mejor trabajo en el futuro, si aquellos mismos casos surgen nuevamente.

Temperatura	Precipitación	Día	Compra	Ropa	Salida.
80	Ninguna	Sábado	No	Casual	Camina
19	Nieve	Lunes	Si	Casual	Maneja
65	Ninguna	Miércoles	No	Casual	Camina
19	Nieve	Lunes	Si	Casual	Maneja

Tabla2.2 Predicción de un nuevo caso

2.5.2 Promediando.

Lidiando con ruido en los datos.

Las situaciones sin embargo, pueden ser más compleja que en el caso anterior. Considere la siguiente situación de la tabla 1.3.

Temperatura	Precipitación	Día	Compra	Ropa	Salida
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	maneja
80	Ninguna	sábado	no	casual	maneja
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina

Tabla2.3 Representación de Datos con ruidos.

Ahora se quiere predecir qué es lo que va a suceder si existe una temperatura de 80 grados, no existe precipitación, si el día de la semana es Sábado, no va de compra y viste ropa casual (Ver tabla 1.4). Se ha visto este caso antes, pero el problema es que ha tenido diferentes respuestas.

Temperatura	Precipitación	Día	Compra	Ropa	
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	maneja
80	Ninguna	sábado	no	casual	maneja
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	?

Tabla2.4 Evaluación de un nuevo caso con ruido.

Hay un par de estrategias posibles para predecir este nuevo caso. Una sería predecir la salida mayoritaria. La vecina camina más veces de lo que maneja en esta situación, por tanto podríamos predecir “camina”. Otra opción es no hacer una predicción sino generar una probabilidad en su lugar, se puede responder que hay una probabilidad de 5/7 que ella caminará.

Temperatura	Precipitación	Día	Compra	Ropa	
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	maneja
80	Ninguna	sábado	no	casual	maneja
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina
80	Ninguna	sábado	no	casual	camina

Tabla2.5 Salida del nuevo caso.

En este estudio, se dará un tratamiento de aprendizaje de máquina que no depende de probabilidades. Pero en muchos casos, los algoritmos y conceptos que trataremos tienen fundamentos probabilísticos y hay muchas más técnicas sofisticadas que dependen de un entendimiento profundo de probabilidades.

Percepción de ruidos.

Lidiando con ruido en los datos.

Aquí está otra situación en la cual el ruido es una cuestión clave. En este momento, el ruido ha alterado nuestra descripción de la situación:

Temperatura	Precipitación	Día	Compra	Ropa	
80	Ninguna	sábado	no	casual	camina
82	Ninguna	sábado	no	casual	camina
78	Ninguna	sábado	no	casual	camina
21	Ninguna	sábado	no	casual	maneja
18	Ninguna	sábado	no	casual	maneja
19	Ninguna	sábado	no	formal	maneja
17	Ninguna	sábado	no	casual	Maneja

Tabla2.6 Lidiando con Ruidos en datos.

En este caso, nos gustaría reconocer que el día que se nos pide hacer la predicción es similar a los casos que hemos visto antes. Es imposible decir si este día es realmente diferente a los otros días. Tendremos que aceptar esa incertidumbre. Por lo tanto, tendremos que tratarlo como una instancia del problema más general que corresponde a la etapa de generalización.

Temperatura	Precipitación	Día	Compra	Ropa	
80	Ninguna	sábado	no	casual	camina
82	Ninguna	sábado	no	casual	camina
78	Ninguna	sábado	no	casual	camina
21	Ninguna	sábado	no	casual	maneja
18	Ninguna	sábado	no	casual	maneja
19	Ninguna	sábado	no	formal	maneja
17	Ninguna	sábado	no	casual	maneja
20	Ninguna	sábado	no	casual	maneja

Tabla2.7 Predicción del nuevo caso

2.5.3 Generalización.

Lidiando con casos no vistos previamente

Observe el siguiente conjunto de datos.

Temperatura	Precipitación	Día	Compra	Ropa	
71	Ninguna	Viernes	Si	Formal	maneja
36	Ninguna	domingo	Si	Casual	camina
62	Llueve	miércoles	No	Casual	camina
93	Ninguna	Lunes	No	Casual	maneja
55	Ninguna	sábado	No	Formal	maneja
80	Ninguna	sábado	No	Casual	camina
19	Nieva	Lunes	Si	Casual	maneja
65	Ninguna	jueves	No	Casual	camina
58	Llueve	Lunes	No	Casual	?

Tabla2.8 Predicción de un nuevo caso.

Ahora considere la siguiente situación: es de 58 grados y llueve un día lunes. La vecina esta vistiendo ropa casual y no necesita ir de compras ¿caminará o manejará? La primera cosa a observar es que no hay una respuesta correcta obvia.

Nunca antes se ha visto este caso. Podríamos no hacer una predicción. Pero, tenemos que decidir. Por tanto, podríamos revisar sobre algunos de los casos acerca dentro del dominio. Podríamos asumir, por ejemplo, que hay una clase de propiedad **suave**: situaciones parecidas se inclinaran a tener categorías parecidas.

Se pueden hacer cualquiera de las siguientes argumentaciones:

1. Ella va a caminar porque está lloviendo hoy y el único día anterior que llovió, ella caminó.
2. Ella va a manejar porque ella ha manejado siempre que es lunes.
3. Ella va a caminar porque ella sólo maneja si viste ropa formal, o si la temperatura está arriba de 90 o debajo de 20.

La pregunta ¿cuál de las tres escoger?, es difícil. Este es uno de los problemas profundos que subyacen en el aprendizaje de máquina.

Imagine que se han dado estos puntos, y se necesita conocer una función de sus coordenadas x, y que tendrá una salida para los puntos rojos y una salida diferente para los puntos negros.

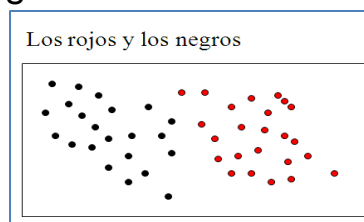


Figura 2.1 Ubicación de Puntos

En este caso, podría ser bastante bueno mediante la definición de una línea que separe las dos clases.

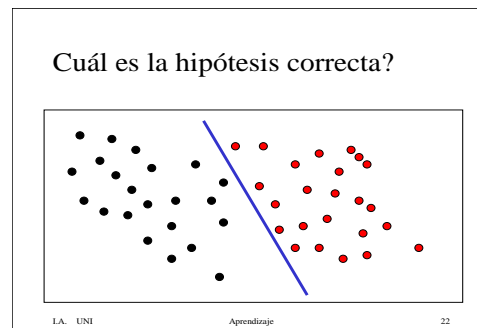


Figura 2.2 Representación de separación de puntos.

Ahora, consideremos la siguiente muestra de punto reflejada en la figura 2.3 No podemos dividirlos con una línea.

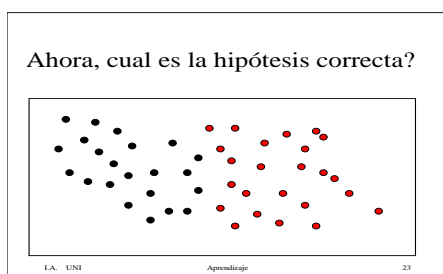


Figura 2.3 Ubicación de puntos un poco dispersos

Pero esta curva como parábola parecería ser un separador razonable.

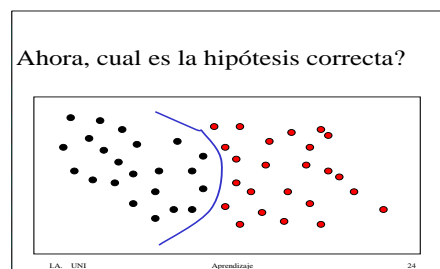


Figura 2.4 Representación de Función separadora

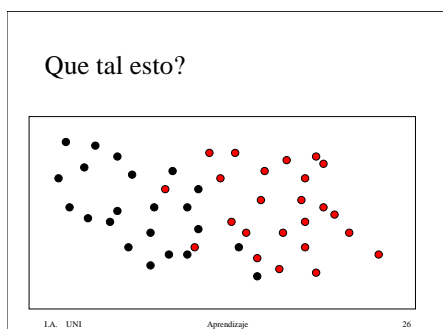


Figura 2.5 Ubicación de puntos más disperso

Ahora observemos la muestra de puntos de la figura 2.5. La separación puede estar dada por varias respuestas diferentes.

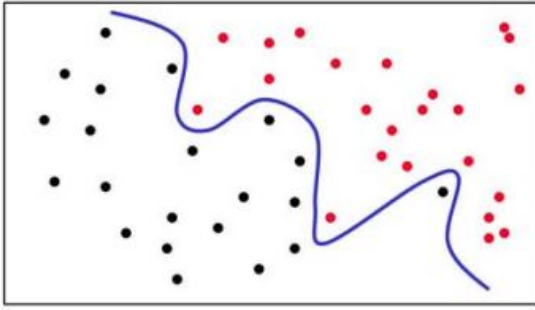


Figura 2.6 Representación de función separadora

Esta podría ser una respuesta: es una separación exitosa de los puntos negros referente a los puntos rojos. Pero sería una función muy complicada.

Aquí esta otra respuesta. No es un separador perfecto, porque se obtienen algunos de los puntos equivocados. Pero sería una función menos compleja.

En general, se enfrenta a tener que sacrificar una cosa con otra, entre hipótesis que cuentan los datos perfectamente y las hipótesis que son, en cierto modo, sencillas.

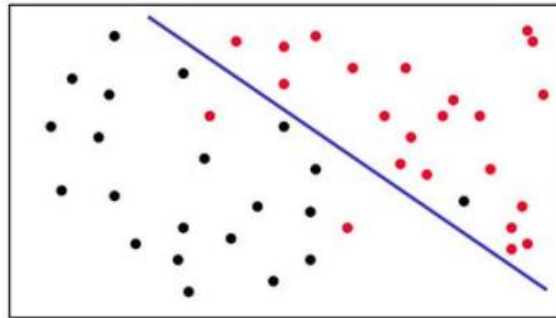


Figura 2.7 Representación de función separadora

2.6 Aprendizaje Supervisado.

Un problema de aprendizaje supervisado esta hecho de un número de requerimiento. Lo primero son los datos (algunas veces llamado conjunto de entrenamiento). Los datos, D , son un conjunto de pares entradas-salidas. Escribiremos el i -avo elemento del conjunto de datos como x^i, y^i . En el contexto de nuestro viejo ejemplo, un x^i es un vector de valores, uno por cada uno de los atributos de entrada, y y^i sería un valor de salida (camina o maneja).

Cada x^i es un vector de n valores. Escribiremos x^i_j para el j -avo atributo de i -avo par de entradas-salidas. Algunas veces nos restringiremos para el caso donde los atributos son solo 0's ó 1's. Otras veces, los dejamos ser seleccionados de un conjunto de elementos discretos (como "nieve", "llueve", "ninguno"). Y, en otros casos, les dejaremos ser valores reales, como temperatura, o peso.

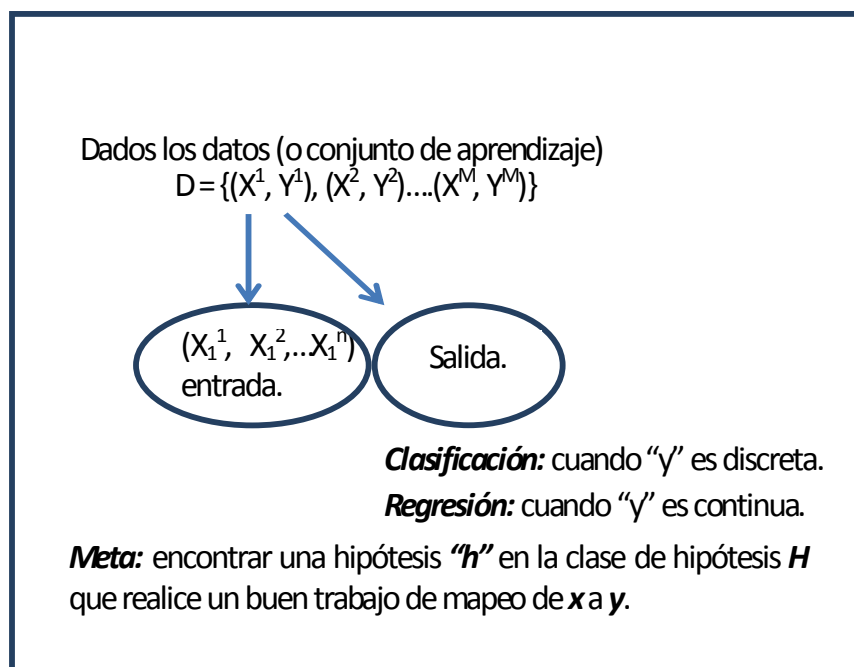


Figura 2.8 Representación de los datos

Similarmente, la salida y^i podría ser booleana, un miembro de un conjunto discreto, o un valor real. Para un problema específico, toda y^i será de un tipo particular.

Cuando y^i es booleano, o un miembro de un conjunto discreto, lo llamaremos un problema de **clasificación**. Cuando y^i es un valor real, lo llamaremos un problema de **regresión**.

Ahora, la meta de aprendizaje será encontrar una hipótesis h , que haga un buen trabajo para indicar la relación entre la entrada y la salida. H es considerada como el conjunto de posibles hipótesis donde nuestro algoritmo elegirá. Esto podría ser algo como un árbol de decisión con 6 nodos, o líneas en un espacio vi -dimensional o una red neuronal con 3 componentes.

2.6.1 La mejor hipótesis.

Así, tenemos un grupo de datos, y una clase de posibles respuestas (hipótesis) y estamos suponiendo regresar el mejor valor de las posibles respuestas. Para hacer esto, necesitamos conocer exactamente qué significa “mejor”. Hay dos típicos componentes para la noción del “mejor”: la hipótesis debe hacer un buen trabajo describiéndolos datos y no debe ser muy complicada.

Particularmente nos gustaría encontrar una hipótesis h tal que, para todos los puntos de datos i , $h(x^i) = y^i$. Aunque no siempre se cumplirá esto, quizás sea verdadero solo para la mayoría de los puntos dados, o la igualdad será debilitada con “no muy lejos de que sean verdaderos y mejores”.

Podemos algunas veces indicar una medida de “error” para una hipótesis de los datos, escribiremos $E(h, D)$. Esto podría ser el número de puntos que son no categorizados por ejemplo.

Otro tema es la complejidad de la hipótesis. Podemos medir la complejidad de un árbol de decisión por el número de nodos que tiene una línea por cuan inclinada sea. En general definiremos una medida de complejidad de la hipótesis en $H, C(h)$.

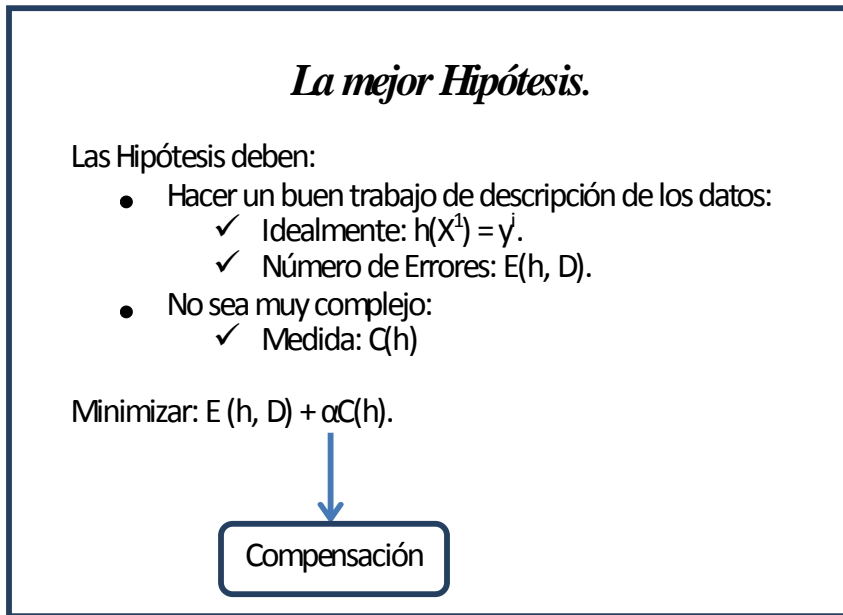


Figura 2.9 Consideración de la mejor hipótesis

¿Por qué nos preocupamos por la complejidad de la hipótesis? Tenemos la intuición que, todas las cosas llegan hacer iguales, las hipótesis más simples son las mejores. ¿Por qué es esto? Hay mucha información teórica y argumentos estadísticos y filosóficos a favor de la simplicidad. Así, por ahora, recurramos a William de Ockham, un teólogo y herético franciscano del siglo 14. Él es famoso por “la Afeitadora de Ockham”, o el principio de parsimonia: ***“non sunt multiplicanda entia praeter necessitatem”*** (***No hay por qué multiplicar entes/cosas a no ser por necesidad***) O sea, ***“las entidades no se multiplican más allá de lo necesario”***. Las personas interpretan esto como que debemos siempre adoptar la hipótesis satisfactoria más simple.

Así dado un conjunto de datos, D , una clase de hipótesis H , una función de ajuste E y una medida de complejidad C , nuestro trabajo será encontrar la h en H que minimice $E(h, D) + \alpha C(h)$, donde α es un número que podemos variar para alcanzar el buen ajuste de los datos versus la complejidad de la hipótesis.

No quisimos, en general, hacer posible esto eficientemente, así, como es usual, tendremos que hacer una cantidad de aproximaciones y atajos. De hecho, la mayoría de los algoritmos de aprendizajes no son descritos exactamente de esta forma. Pero es el principio subyacente de lo que hacemos.

2.6.2 Aprendizaje como búsqueda.

¿Cómo podemos encontrar la hipótesis con el menor valor de nuestro criterio establecido? Podemos certeramente pensar en esto como un problema de búsqueda, aunque muchos algoritmos de aprendizaje no parecieran hacer búsqueda. Para alguna combinación cuidadosamente escogida de clases de hipótesis y criterio de error, podemos calcular la hipótesis óptima directamente. En otros casos, el espacio de hipótesis puede ser descrito como un vector de parámetros de valores reales, pero por los cuales no podemos calcular el valor óptimo. Una práctica común de estos problemas es hacer una clase de búsqueda local escalada llamada ascenso gradiente (o descenso).

Aprendizaje como Búsqueda.

¿Cómo podemos encontrar la hipótesis con el valor más pequeño de $E(h, D) + \alpha C(h)$? Buscándolo!

- ✓ Para algunas clases de hipótesis podemos calcular la dirección óptima de ***h*** directamente! (separadores lineales).
- ✓ Para otros, se pueden hacer búsquedas locales (gradiente descendente en redes neuronales).
- ✓ Para algunos espacios estructurados construir la hipótesis vorazmente.

Figura 2.10 Aprendizaje como búsqueda

Finalmente, algunas veces es posible construir una hipótesis iterativamente, empezando con una hipótesis simple y agregarle para ajustarse mejor a los datos. Algunos métodos son frecuentemente llamados “voraces”, porque ellos comienzan tratando de tomar la mejor hipótesis simple, y le sumamos la otra mejor encontrada, etc. El resultado no es el más óptimo, pero algunas veces es muy bueno. Comenzaremos por revisar algunos métodos de esta clase.

2.6.3 Validación cruzada.

Algunas veces le gustaría conocer que tan bien un algoritmo de aprendizaje particular trabaja, en vez de evaluar lo que una única hipótesis habría producido. Corriendo el algoritmo solo una vez y evaluando su hipótesis solo le da un ejemplo de su comportamiento, y podría ser difícil decir que tan bien realmente trabaja solo con un ejemplo.

Si se tiene una gran cantidad de datos, puede dividirlos en grupos, y usar la mitad de sus grupos individuales para entrenar la nueva hipótesis y la otra mitad de

ellos para evaluar la hipótesis. Pero raramente tendrá suficientes datos para hacer esto.

Una solución es llamada validación-cruzada. Se dividen los datos en algún número (digamos por ahora 10) pedazos.

Validación Cruzada.

- Para evaluar el desempeño de un algoritmo como un todo (en vez de una hipótesis particular):
 - Divida los datos en K subconjuntos.
 - Realizar K veces.
 - Entrene con $k - 1$ subconjuntos.
 - Evalúe con el subconjunto apartado.
- Regrese el promedio de un puntaje al realizar las K evaluaciones.
- Método útil para predecir qué clase de algoritmo usar sobre un conjunto de datos particulares (no para evaluar la calidad de una hipótesis particular).

Figura 2.11 Utilizando validación cruzada

Comience alimentando con los trozos del 1 al 9 a su sistema de aprendizaje y evalúe las hipótesis resultantes con el trozo 10. Ahora, ya terminó de alimentar al sistema de aprendizaje con el pedazo 9 y evalúe la hipótesis resultante del trozo 9. Y así en adelante con el 8, 7...

Al final de este proceso, usted habrá corrido el algoritmo de aprendizaje 10 veces, obteniendo 10 diferentes hipótesis y evaluando cada una de ellas sobre un conjunto de prueba diferente. Puede ahora promediar resultados de cada uno de estos ensayos para obtener un imagen global de qué tan bien su algoritmo de aprendizaje se desempeña. (Para aquellos de ustedes que pueden estar estadísticamente sofisticados, tienen que ser cuidadosos con estas estadísticas porque los ejemplos no son independientes.) Recuerden que este es un método para comparar algoritmos sobre un conjunto de datos, no para evaluar la calidad de una hipótesis particular.

Curva de aprendizaje.

Una cosa interesante de ver es que tan bien trabaja el proceso de aprendizaje como una función de las cantidades de datos dispone. Hice esta curva de

aprendizaje por escoger una función meta para generar los datos, dibujando puntos de entradas uniformemente por medio de random y generando sus asociadas y usando mi función secreta. Primero genere 10 puntos y entrené una hipótesis con ellos, entonces la evalué. Entonces lo hice para 20 puntos, y así en adelante.

Esta es la curva de aprendizaje para una función bastante simple: $f_{344} \wedge f_{991} \vee f_{103} \wedge f_{775}$, en un dominio con 1000 atributos. El eje x es la cantidad de datos de entrenamiento. El eje y es el numero de errores hechos por la hipótesis resultante, sobre un conjunto de prueba de datos que se dibujó uniformemente con random, como un porcentaje del tamaño total del conjunto de entrenamiento.

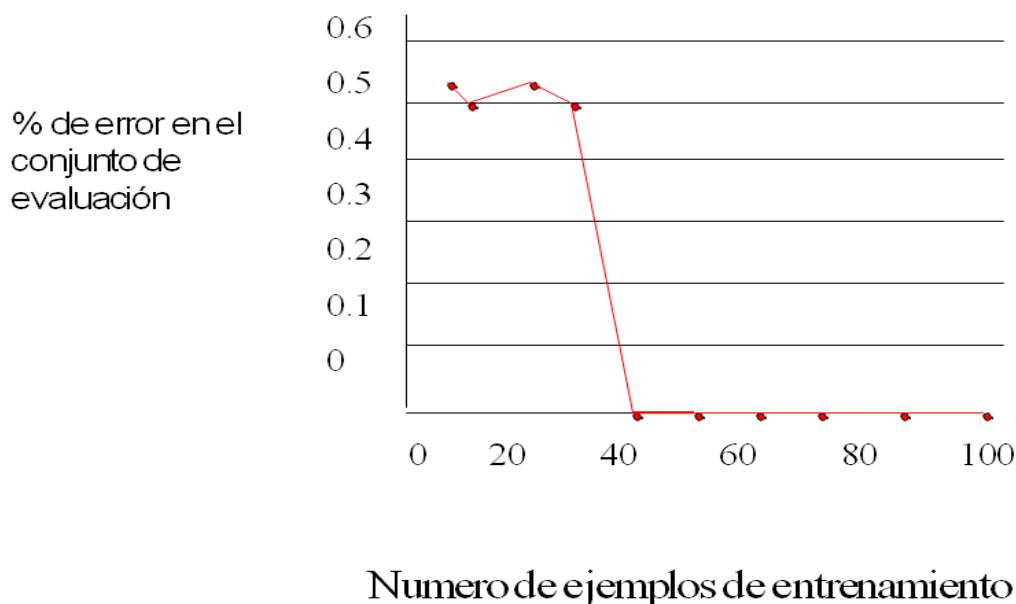


Figura 2.12 Representación de errores en validación cruzada.

Esto toma sorprendentemente pocos ejemplos antes de encontrar la hipótesis correcta. Con 1000 características, hay 2^{1000} ejemplos posibles. Estamos obteniendo la hipótesis correcta después de ver una pequeñísima porción del espacio completo.

Métodos de Aprendizajes

Cap. 3

3. Métodos de Aprendizaje.

Los métodos de aprendizaje se pueden clasificar en términos de:

1. La forma de la hipótesis
2. La manera en que la computadora encuentra una hipótesis dado unos datos

Ambas clasificaciones de aprendizaje difieren en términos de clases de hipótesis con que trabajan y los algoritmos que usan para encontrar una buena hipótesis dado los datos.

3.2 Vecino más cercano (LazyLearning).

Uno de los algoritmos de aprendizaje es el llamado vecino más cercano (o lazylearning).

El mecanismo básico con el que trabaja la técnica de vecino más cercano es la siguiente

1. Recuerda todos sus datos
2. Cuando alguien realiza una pregunta:
 1. Encuentra el punto de datos más cercano.
 2. Regresa la respuesta asociada a éste caso

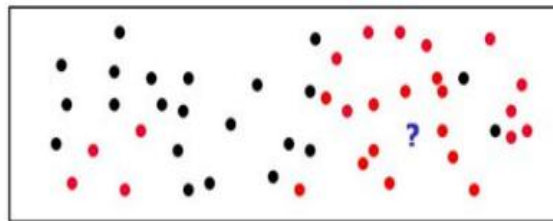


Figura 3.1 Ubicando un nuevo caso

Así, en este caso, el punto más cercano a la pregunta es rojo, así retornamos rojo.

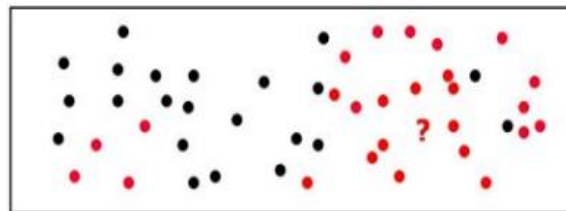


Figura 3.2 Clasificación de un nuevo caso.

Mecanismo de trabajo:

Consideremos un conjunto de datos trabajo.

F ₁	F ₂	F ₃	F ₄	y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	1	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0

Tabla 3.1 Conjunto de Entrenamiento

Para este caso realizaremos una transformación booleana, y de acuerdo al nuevo caso vemos los casos vecinos es decir los casos más parecidos de acuerdo a su entrada.

Por ejemplo consideremos el nuevo caso <1 0 1 1>, los casos vecinos de acuerdo a los valores de los atributos, el casos más parecido es caso en que los atributos tienen el valor de <1 0 0 1> que contienen 3 atributos con el mismo valor del nuevo caso. Véase Figura en la cual se muestra de forma ordenada los casos más parecido.

F1	F2	F3	F4	y	
1	0	0	1	0	3
1	1	1	1	0	3
0	0	1	1	1	3
1	0	1	0	1	3
0	1	1	1	1	2
1	1	0	1	0	2
1	0	0	0	0	2
0	1	1	0	1	1
0	0	0	0	1	1

Tabla 3.2 Obteniendo casos vecinos

Aquí podemos ver que hay casos vecinos que coinciden en al menos 3 de los valores de los atributos del nuevo caso. Si consideramos que el caso sea evaluado con su primer vecino más cercano coincidimos la salida con el caso vecino en este caso la salida para el nuevo caso 0. Si consideramos los 3 casos

vecinos la salida será la mayoritaria a sus casos vecinos en éste caso 0, pero si consideramos los 4 vecinos más cercano la salida al nuevo caso puede ser cualquiera o respondemos que no es posible dar una respuesta a este caso, a menos que se dé una razón probabilística.

Sin embargo el método tiene la flexibilidad de que la transformación de nuestros datos no sea únicamente booleana, trataremos en este caso transformación booleana, el método de vecino más cercano trabaja de 2 modos principales:

- ✓ Vecino más cercano, haciendo una transformación de datos.
- ✓ Distancia Manhattan, que no requiere transformación.

Este último, que para casos numéricos no se realizan ninguna transformación pues su mecanismo básico de trabajo consiste en calcular las distancia que hay entre el nuevo caso y el atributo, esto se calcula para todos los casos, luego ordenamos los casos de menor distancia a mayor y luego se considera la salida de acuerdo a la cantidad de vecinos a considerar.

3.3 Árboles de Decisión.

Otro algoritmo de aprendizaje construye hipótesis en la forma de árboles de decisión. En un árbol de decisión, cada nodo representa una pregunta, y los arcos representan posibles respuestas. Podemos usar este árbol de decisión para encontrar que predicción debemos hacer en el problema de maneja/camina.

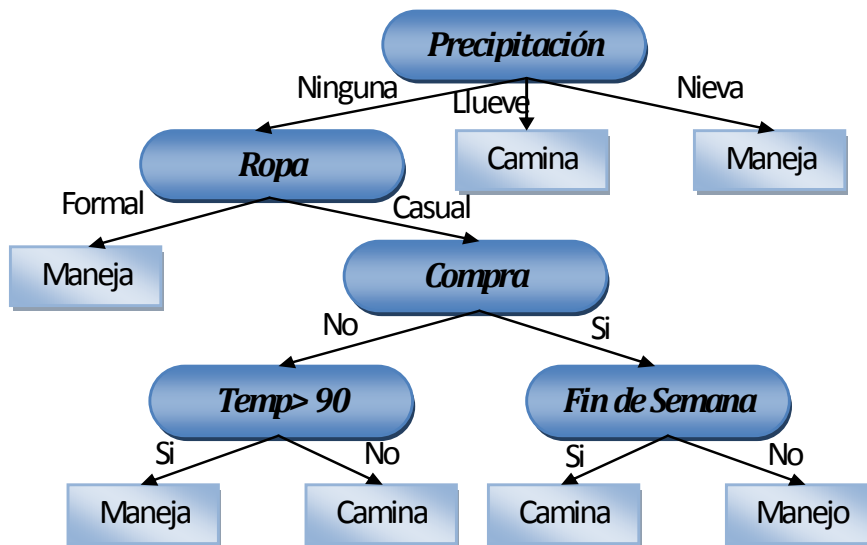


Figura 3.3 Árbol de decisión

Iniciamos preguntando cual es la precipitación actual. Si nieva, paramos de hacer preguntas y se tiene como salida maneja.

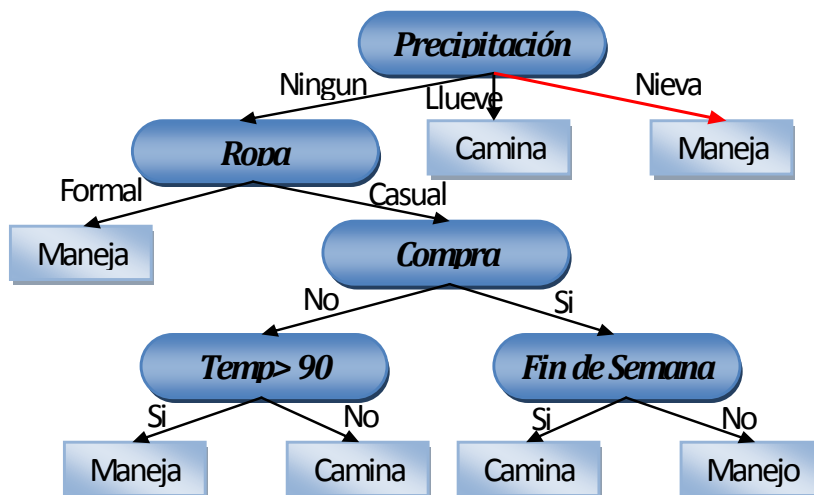


Figura 3.4 Obteniendo salida para un nuevo caso.

Si no hay precipitaciones, entonces tenemos que preguntar qué clase de ropa lleva la vecina

Si viste formalmente, ella manejará, sino, tenemos que hacer otra pregunta, hasta que tengamos una salida.

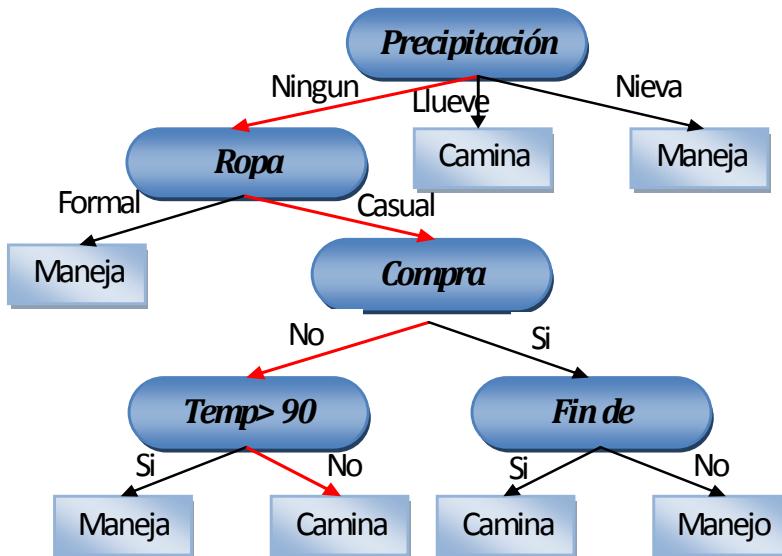


Figura 3.5 Obteniendo salida para un nuevo caso.

Hipótesis como ésta son buenas por una variedad de razones una de ellas y de vital importancia es que resultan relativamente fáciles de interpretar por humanos. Así, en algunos casos, corremos un algoritmo de aprendizaje con algunos datos y entonces mostramos el resultado a un experto en el área (astrónomos, físicos), y ellos encuentran que el algoritmo de aprendizaje ha encontrado algunas regularidades en sus datos que son de interés real para ellos.

Los métodos de aprendizaje de máquina han sido exitosamente utilizados en una gran variedad de aplicaciones, incluyendo:

3. Asesorando el riesgo de un préstamo crediticio
4. Detectando fraudes con tarjetas de crédito
5. Catalogando imágenes astronómicas
6. Detectando y diagnosticando fallas de fabricación
7. Personalizando noticias y búsqueda en la WEB
8. Conduciendo un carro automáticamente a través de EEUU

3.3.1 Árboles de decisión y aprendizaje Automatizado.

Para trabajar con esta técnica se considera los atributos de entrada y su correspondiente salida en la cual la primera tarea es convertir dichos datos en datos booleanos.

La clase de hipótesis será un árbol de decisión, análogo. Un árbol de decisión es más que un árbol como el que se acaba de considerar en la sección anterior en cada nodo interno (no hoja), que corresponderá a un atributo de nuestro conjunto de datos. En árbol de aprendizaje hay dos arcos saliendo de cada nodo, y de acuerdo a las consideraciones serán etiquetado 0 o 1, representando los dos posibles valores que los atributos pueden tomar. Teniendo en cuenta que únicamente trabajamos con datos booleanos. Ver figura 3.6.

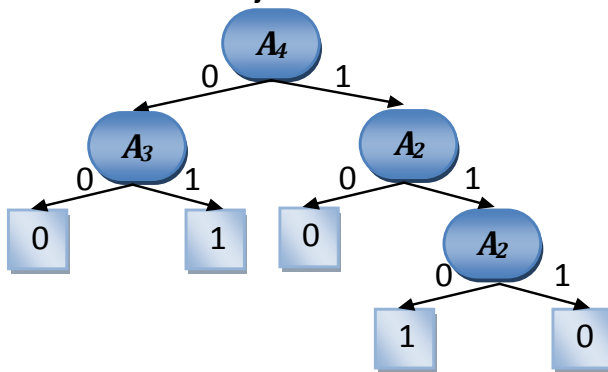


Figura 3.6 Árbol de decisión binario

Los árboles representan funciones Booleanas de x 's (vectores de atributos) en datos Booleanos (p.e [1 0 1 1]). Para calcular la salida para un vector de entrada x^i , comenzamos en la raíz del árbol. Observamos el atributo que está como nodo raíz, es el atributo j , y vemos cual es el valor en el vector de ese mismo atributo x_j^i . Luego bajamos el arco correspondiente a ese valor. Si llegamos a otro nodo interno, revisamos el valor del atributo, siguiendo el arco correcto, y así continuamos hasta llegar a un nodo hoja del árbol. Cuando lleguemos a un nodo hoja, tomamos la etiqueta que encontramos y la tomamos como salida.

Consideremos como un ejemplo, la entrada [0 1 1 0] generara una salida de 0 (porque el tercer elemento de la entrada tiene valor 1 y el primero tiene valor 0, lo cual toma la hoja etiquetada 0).

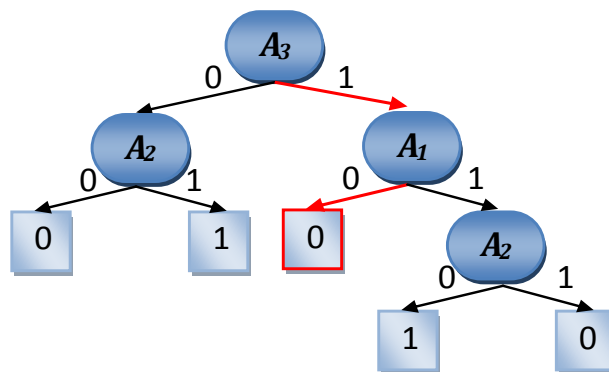


Figura 3.7 Evaluando un nuevo caso

Los árboles de decisión son una forma de representar expresiones Booleanas. Para convertir un árbol a una expresión se consideran las ramas donde se llega a los nodos hojas 1 y cada atributo es unido mediante una conjunción, es decir se describe la condición que tuvo que tener la entrada en el orden para que haya obtenido esa hoja del árbol. Finalmente se toman todas las conjunciones que están generando la salida 1 y se unen mediante una disyunción.

Por lo tanto, para nuestro ejemplo la expresión que describe nuestro árbol es $(\neg$

Si permitimos negaciones de atributos primitivos en forma normal disyuntiva, entonces ambos, forma normal disyuntiva y árboles de decisión están capacitados para representar cualquier función booleana. ¿Por qué, entonces, nos deberíamos molestar en cambiar representaciones de esta manera? La respuesta es que vamos a definir nuestra tendencia en una forma que es natural para arboles, y usamos un algoritmo que es confeccionado para funciones de aprendizaje en la representación en árbol.

La aplicación de la navaja de Ockham nos lleva a preferir árboles que son pequeños, medidos por el número de nodos. Como antes, sería imposible de calcular el árbol del mínimo tamaño que satisface nuestro criterio de error. Como antes, deberíamos ser cuidadoso, creciendo el árbol en una manera que parece esto sería lo mejor en cada paso

Consideraremos un par de métodos para hacer seguro que nuestro árbol resultante no son muy largos, así podemos evitar sobre ajustes.

Es interesante ver que un árbol de decisión pequeño no corresponde a que su expresión en FND también lo sea. Veamos el siguiente ejemplo en el cual su FND es muy compleja junto con la expresión FND resultante.

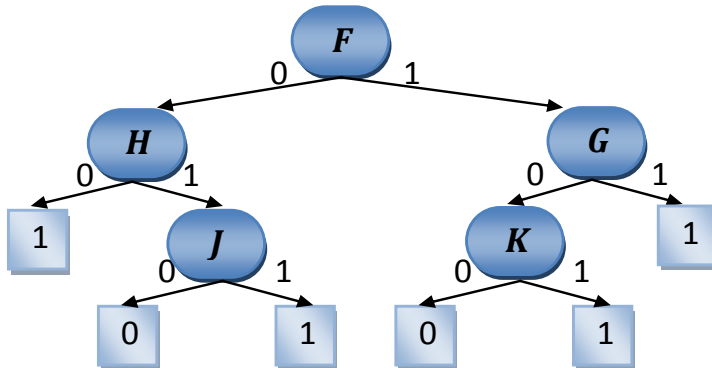


Figura 3.8 Árbol de decisión y FND

Pero aquí está un caso en el cual una expresión muy simple FND requiere un árbol grande para representarlo. No hay una razón particular para preferir árboles sobre FND o FND sobre árboles como una clase de hipótesis.

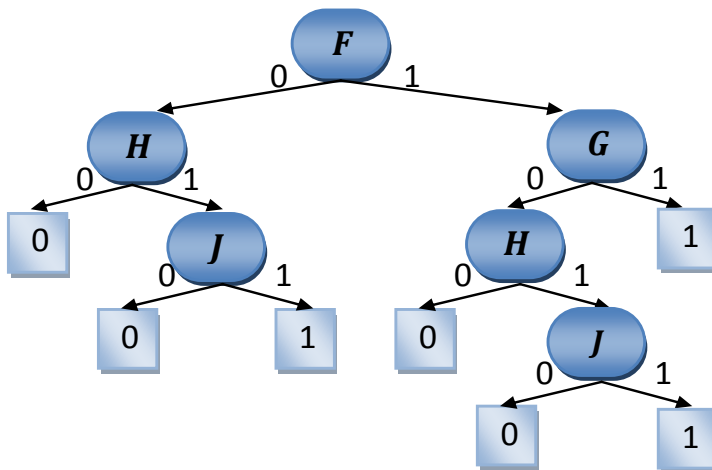


Figura 3.9 FND y árbol de decisión

3.3.2 Algoritmo para Construcción de árboles de decisión

La idea del aprendizaje por árboles de decisión y el algoritmo para hacerlo fue, desarrollado por investigadores independientes en estadística (Breiman, Friedman, Olsen y Stone) e investigadores en I.A (Quinlan), en el periodo aproximado de 1980.

El árbol se construye de arriba hacia abajo. A continuación se muestra el pseudocódigo para el algoritmo. El cual toma como entrada un conjunto de datos y su salida será el árbol.

ConstruyaArbol (Datos)

Si todos los elementos de Datos tiene el mismo valor y, entonces

Hacer-nodo-hoja (y)

De lo contrario

Atributo:= tomeMejorAtributo (Datos)

HagaNodoInterno (atributo,

ConstruyaArbol (SeleccionFalso (Datos, atributo)),

ConstruyaArbol (SeleccionVerdad (Datos, atributo)))

Primero evaluamos para ver si todos los elementos de los datos tienen el mismo valor **y**. si es así, simplemente haremos un nodo hoja con ese valor **y** y ya está hecho. Este es el caso base de nuestro algoritmo recursivo.

Si tenemos una mezcla diferente de valores **y**, escogemos un atributo que será nuestro nuevo nodo del árbol. Entonces dividimos los datos en dos conjuntos, aquellos para los cuales el valor de **y** del atributo es 0 y aquellos para los cuales el valor **y** es 1. Finalmente, llamamos al algoritmo recursivamente sobre cada uno de estos conjuntos de datos. Usamos el atributo seleccionado y los dos sub árboles creados recursivamente para construir nuestro nuevo nodo interno.

3.3.3 Escogiendo el mejor atributo.

Nuestra meta para construir el árbol, por lo que tenemos que escoger un atributo, de tal forma hacer 2 subconjuntos en donde el atributo considerado es cero y el otro subconjunto donde el atributo es uno, luego por cada subconjunto volvemos a seleccionar y a subdividir

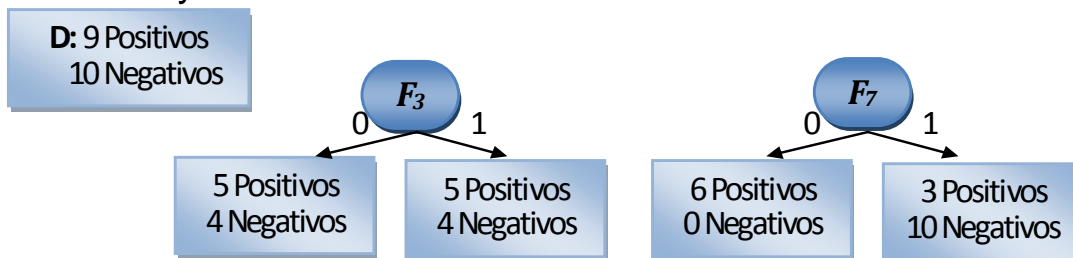


Figura 3.10 Separación de casos para cálculo de entropía.

Consideremos la siguiente situación, en la cual separamos la cantidad de casos positivos y casos negativos del total de conjunto de datos y luego una separación de por cada atributo

En este ejemplo, separar en base a F_7 es de gran ayuda. Para formalizar esa intuición, necesitamos desarrollar una medida del grado de uniformidad por cada subconjunto de los datos, y de esta forma seleccionar el mejor atributo.

Comenzaremos buscando una medida estándar de desorden, usada en física y teoría de la información llamada **entropía**. Solamente consideramos esto en el caso binario, por ahora.

Sea p la proporción de ejemplos positivos en un conjunto de datos (o sea, el número de ejemplos positivos divididos por el número total de ejemplos). Entonces la entropía de este conjunto de datos por la siguiente fórmula:

La siguiente gráfica corresponde a una representación de la entropía en función de p . Cuando p es 0 o 1, entonces la entropía es 0. Tenemos que ser un poco cuidadosos, cuando p es 1 entonces $1 \log 1$ es claramente 0. Pero cuando p es 0 $\log 0$ es infinito negativo. Pero consideraremos 0, por tanto $0 \log 0$ es también 0.

p : proporción de los ejemplos positivos en un conjunto de datos

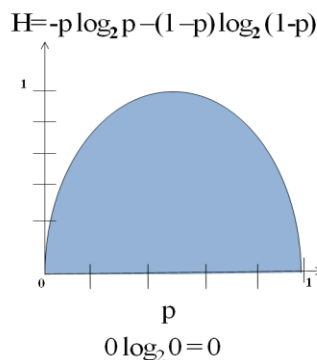


Figura 3.11 Representación de la Entropía

Así, cuando todos los elementos del conjunto tienen el mismo valor en relación a su salida, ya sea 0 o 1, entonces la entropía es 0. No hay desorden.

La función entropía es maximizada cuando $p=0.5$. Cuando p es la mitad, el conjunto está desordenado al máximo. No hay bases para suponer cual será la respuesta.

Cuando separamos los datos por el atributo j , obtenemos dos conjuntos de datos. Llamaremos al conjunto de ejemplos para cada atributo j con valor 1 D_j^+ y aquellos para los cuales j tiene un valor de 0 D_j^- .

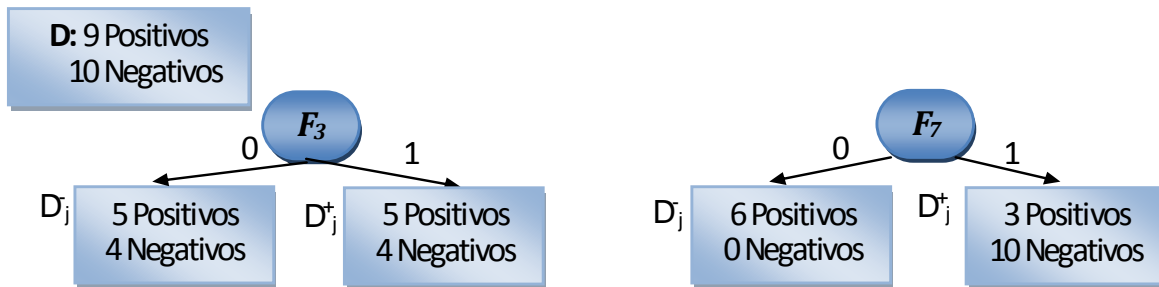


Figura 3.12 Conjunto de ejemplos positivos y negativos por atributo

Podemos calcular la entropía para cada uno de estos subconjuntos y se considera la siguiente notación:

$H(D_j^+)$ para el cálculo de la entropía en la cual el atributo j tiene valor 1 y

$H(D_j)$ para el cálculo de la entropía en la cual el atributo j tiene valor 0

Ahora, tenemos que encontrar la manera de combinar estos dos valores de entropías podríamos sumarlos o promediarlos. ¿Pero qué pasa si tenemos la siguiente situación; en la cual hay un caso con salida positiva en un conjunto de datos y 100 casos positivos y 100 casos negativos? Podríamos considerar que no se ha hecho una buena separación, pero si promediamos la entropía, obtendríamos un valor de $\frac{1}{4}$.

Usaremos un promedio de pesos para combinar la entropía de los dos conjuntos. Consideraremos P_j como la proporción de ejemplos en el conjunto de datos D para los cuales el atributo j tiene valor 1. Calcularemos un peso entropía promedio para separar sobre el atributo j como

$$AE(j) = p_j H(D_j^+) + (1-p_j) H(D_j)$$

Por ejemplo podemos ver que la separación con el atributo F_7 es mucho más útil, según lo reflejado en la entropía promedio por cada separación.

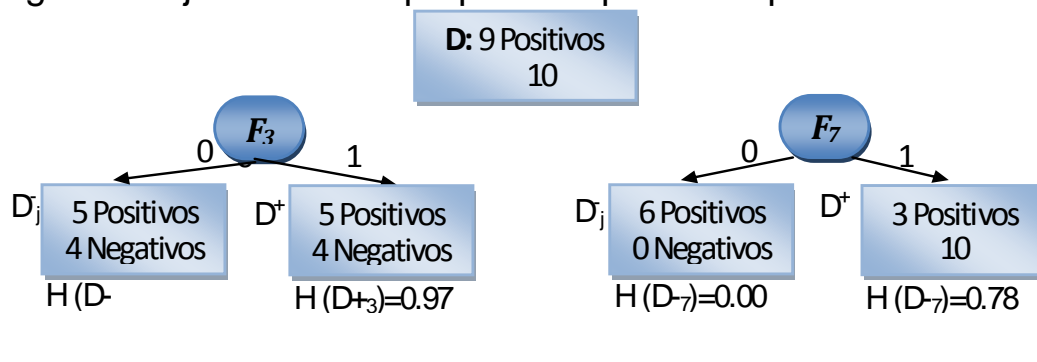


Figura 3.13 Cálculo de entropía promedio

Regresando a nuestro algoritmo, entonces seleccionaremos el atributo en cada paso que reduzca al mínimo la entropía promedio.

Como es usual, cuando hay ruido en los datos, es fácil ajustarlo. Podría ser concebible crecer el árbol al punto donde hay un sólo dato en cada nodo hoja. (O tal vez no: de hecho, si tenemos dos puntos de datos con el mismo valor x pero valores de y diferentes, nuestro algoritmo actual nunca terminaría, lo cual es ciertamente un problema). Por tanto, al menos, tenemos que incluir una evaluación para asegurarnos que haya una separación que hace ambos subconjuntos de datos no vacíos. Si no hay, no tenemos opción, pero paramos y hacemos un nodo hoja.

¿Qué debemos hacer si tenemos que parar y hacer un nodo hoja cuando los puntos de datos en el nodo tienen diferentes valores y ? Escogiendo el valor mayoritario y es la mejor estrategia. Si aquí hay un número igual de puntos negativos y positivos, entonces tome un y arbitrario.

Nos detenemos cuando se tienen múltiples casos con la misma x y con diferentes valores de y . Hacemos el nodo hoja con la salida que predomina en los valores de y . Consideraremos para parar para evitar un sobre ajuste cuando la entropía de un conjunto de datos está debajo de cierto umbral y cuando el número de elementos de un conjunto de datos está debajo del umbral y cuando la entropía promedio no disminuye significativamente.

Pero en el crecimiento del árbol, puede en algunas veces resultar un sobreajuste. La solución más simple es cambiar la evaluación en el caso base haciendo un umbral con la entropía. Si la entropía está bajo de algún valor ϵ , decidimos que esta hoja está muy cercana a lo puro.

Otra solución simple es tener un umbral sobre el tamaño de sus hojas; si el conjunto de datos de alguna hoja tiene menos que el número de elementos, entonces no se divide aún más.

Otro método posible es solo separar si la separación representa una mejora real. Podemos comparar la entropía del nodo actual con el promedio de la entropía para el mejor atributo. Si la entropía no se disminuye significativamente, entonces podríamos renunciar

3.3.4 Simulación del Algoritmo.

Consideremos el siguiente conjunto de datos, también llamado conjunto de entrenamiento. Nuestro primer paso es calcular las entropías promedios por cada atributo.

F ₁	F ₂	F ₃	F ₄	y	Nuestro conjunto de datos tiene una entropía de $H(D)=0.92$ La entropía promedio por cada atributo son las siguientes $AE_1=0.92$ $AE_2=0.92$ $AE_3=0.81$ $AE_4=1$
0	1	1	0	0	
1	0	1	1	1	
1	1	1	0	1	
0	0	1	1	1	
1	0	0	1	0	
0	1	1	1	0	

Tabla 3.3 Conjunto de Entrenamiento y entropías promedio

En este caso se puede observar que el mejor atributo para separar es el atributo F_3 dado que es el que tiene menos desorden en sus datos

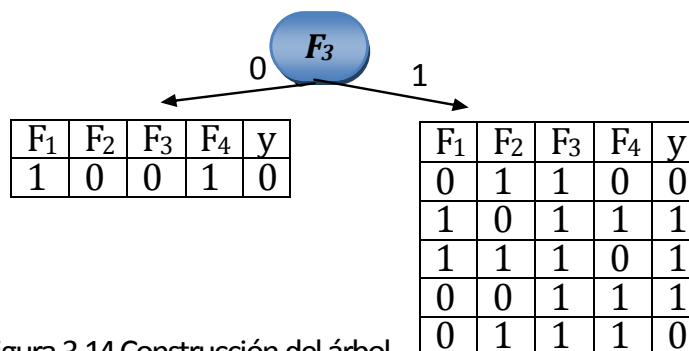


Figura 3.14 Construcción del árbol

Podemos observar que al hacer la separación en los casos donde el atributo 3 tiene valor 0 resulta un único caso por tanto hacemos nodo hoja con el valor correspondiente a la salida que corresponde a cero. Para los casos en donde el atributo 3 tiene valor 1 vemos otro comportamiento, por lo que es necesario hacer el cálculo nuevamente por cada atributo a excepción del nodo considerado.

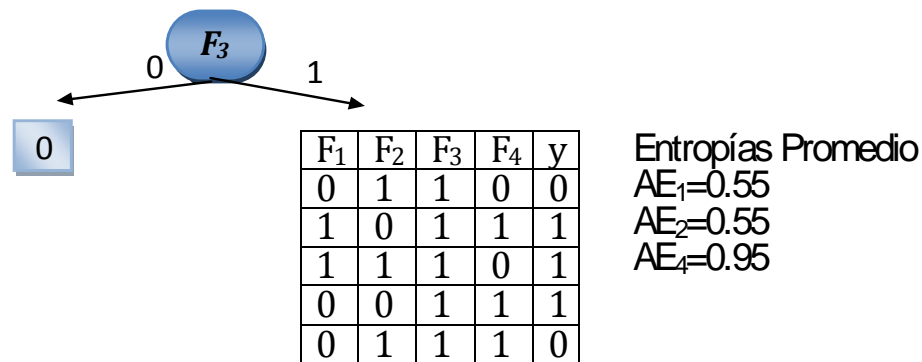


Figura 3.15 Construcción de árbol de aprendizaje

Siguiendo el algoritmo nuevamente seleccionamos el atributo con menor entropía promedio, podemos observar que el atributo 1 y el atributo 2 tienen la misma entropía por lo cual podemos elegir arbitrariamente. Seleccionaremos el atributo 1 y hacemos la separación correspondiente al atributo.

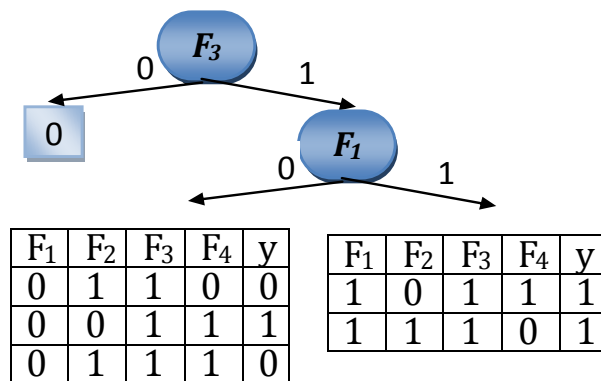


Figura 3.16 Separación de casos por el atributo 1

Cuando separamos vemos que cuando el atributo 1 tiene valor 1 sus salidas también tienen salida un, por tanto hacemos como nodo hoja 1. Cuando el atributo 1 tiene valor 0 es necesario calcular nuevamente la entropía promedio para los atributos. Haciendo este cálculo el atributo con menor entropía es el atributo 2, y hacemos su separación respectiva.

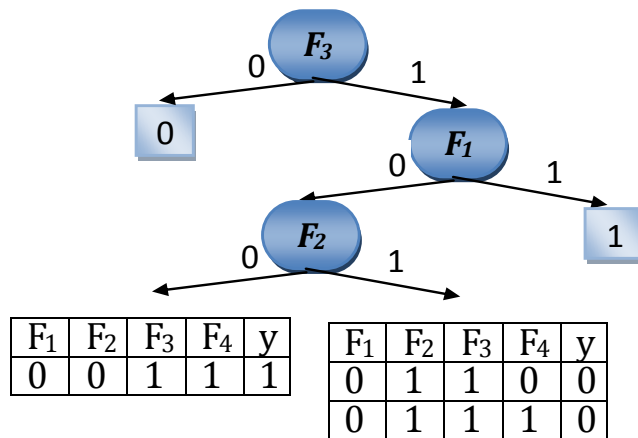


Figura 3.17 Separación de casos por el atributo 2

Cuando hacemos la separación correspondiente al atributo 2, vemos que en ambos lados las salidas son iguales, hacemos los nodos hojas correspondientes a la salida. Esto nos conllevaría a que el algoritmo termine y el árbol.

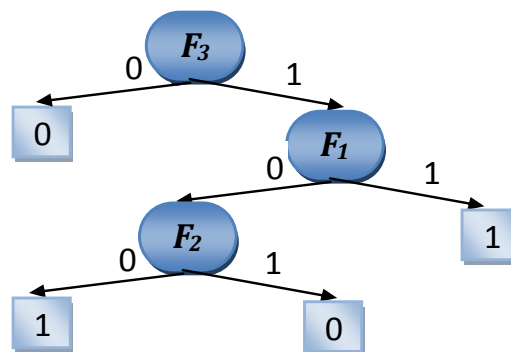


Figura 3.18 Árbol resultante

3.4 Naive Bayes.

Naive Bayes es otro algoritmo de aprendizaje automatizado, creado por el reverendo Thomas Bayes, quien desarrolló un importante razonamiento de teoría probabilística.

El ampliamente usado in aplicaciones con grandes cantidades de atributo. Sus características principales son:

1. Basada en la regla de Bayes para inferencia probabilística
2. Probabilidad de hipótesis basada en evidencia.
3. Selecciona la hipótesis con el máximo de probabilidades después que la evidencia ha sido incorporada.
4. El algoritmo es particularmente útil para dominios con muchísimos atributos.

Consideremos el siguiente conjunto de datos:

F ₁	F ₂	F ₃	F ₄	y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

Tabla 3.4 Conjunto de Entrenamiento para aplicar Naive Bayes.

El primer paso es caracterizar los atributos individualmente de acuerdo a cada ejemplo. Lo primero es observar los ejemplos positivos y vemos que fracción de ejemplos positivos existen cuando el atributo tiene valor 1 y que fracción de ejemplos positivos existen cuando el atributo tiene valor 0.

F ₁	F ₂	F ₃	F ₄	Y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

Tabla 3.5 Conjunto de entrenamiento.

De igual manera, posteriormente observamos los ejemplos negativos y vemos que fracción ejemplos negativos hay cuando el atributo tiene valor 0 y que fracción de ejemplos negativos hay cuando el atributo tiene valor 1.

F ₁	F ₂	F ₃	F ₄	Y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

R₁ (1,1)=1/5: Fracción de todos los ejemplos positivos cuando el atributo 1 tiene valor 1.

R₁ (0,1)=4/5: Fracción de todos los ejemplos positivos cuando el atributo 1 tiene valor 0.

R₁ (1,0)=5/5: Fracción de todos los ejemplos negativos cuando el atributo 1 tiene valor 1.

R₁ (0,0)=0/5: Fracción de todos los ejemplos negativos cuando el atributo 1 tiene valor 0.

Tabla 3.6 Conjunto de entrenamiento y evaluación por atributos

De la misma manera realizamos el cálculo para cada atributo, como se muestra en la figura.

F ₁	F ₂	F ₃	F ₄	Y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

R₁(1,1)=1/5 R₁(0,1)=4/5

R₁(1,0)=5/5 R₁(0,0)=0/5

R₂(1,1)=1/5 R₂(0,1)=4/5

R₂(1,0)=2/5 R₂(0,0)=3/5

R₃(1,1)=4/5 R₃(0,1)=1/5

R₃(1,0)=1/5 R₃(0,0)=4/5

R₄(1,1)=2/5 R₄(0,1)=3/5

R₄(1,0)=4/5 R₄(0,0)=1/5

Tabla 3.7 Conjunto de Entrenamiento y fracciones probabilísticas

Estos conjunto de fracciones representan nuestra hipótesis, a diferencia de de la técnicas de árboles de decisión que su hipótesis es un árbol de decisión. Ahora

dato un nuevo caso $x = \langle 0 \ 1 \ 0 \ 0 \rangle$ podemos utilizar ésta hipótesis para calcular el valor de su salida y

$$\begin{aligned} R_1(1,1) &= 1/5 & R_1(0,1) &= 4/5 \\ R_1(1,0) &= 5/5 & R_1(0,0) &= 0/5 \\ R_2(1,1) &= 1/5 & R_2(0,1) &= 4/5 \\ R_2(1,0) &= 2/5 & R_2(0,0) &= 3/5 \end{aligned}$$

$$\begin{aligned} R_3(1,1) &= 4/5 & R_3(0,1) &= 1/5 \\ R_3(1,0) &= 1/5 & R_3(0,0) &= 4/5 \\ R_4(1,1) &= 2/5 & R_4(0,1) &= 3/5 \\ R_4(1,0) &= 4/5 & R_4(0,0) &= 1/5 \end{aligned}$$

Consideremos el siguiente ejemplo $x = \langle 0, 0, 1, 1 \rangle$, para saber el resultado de este caso realizaremos 2 cálculos, uno correspondiente para los ejemplos positivos y otro cálculo para los ejemplos negativos.

El primer cálculo se realiza en relación a los ejemplos positivos, y de acuerdo a los valores de cada atributo del nuevo caso. Para nuestro ejemplo el atributo uno tiene valor 0 entonces usamos de nuestra hipótesis R_1 de 0, 1, para el atributo 2 igual a cero, usamos R_2 de 0, 1, para el atributo 3 igual 1, usamos R_3 de 1, 1 y para el atributo 4 igual a 1, usamos R_4 de 1, 1.

Cada factor en la respuesta representa el grado el cual este atributo tiende a tener este valor en lo ejemplos positivos, multiplicado todas juntas nos da una medida de que el nuevo caso tenga una salida positiva.

F ₁	F ₂	F ₃	F ₄	Y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

Nuevo Caso $x = \langle 0, 0, 1, 1 \rangle$

$$S(1) = R_1(0,1) * R_2(0,1) * R_3(1,1) * R_4(1,1)$$

$$S(1) = (4/5) * (4/5) * (4/5) * (2/5) = 0.205$$

Tabla 3.8 Evaluación del nuevo caso utilizando Naive Bayes.

Hacemos los mismo pero ahora tomando en cuenta los casos negativos, obteniendo un resultado para x siendo un caso negativo. Algo muy diferente sucede aquí, porque cuando el atributo 1 es cero en R_1 de 0, 0, no existe en el conjunto de casos un caso así, de aquí se deduce que cuando el caso x nunca será negativo.

F ₁	F ₂	F ₃	F ₄	Y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

Nuevo Caso x=<0, 0, 1, 1>

$$S(0) = R_1(0,0) \cdot R_2(0,0) \cdot R_3(1,0) \cdot R_4(1,0)$$

$$S(0) = (0) \cdot (3/5) \cdot (4/5) \cdot (2/5) = 0$$

Tabla 3.9 Evaluación del nuevo caso.

Una vez que tengamos los resultados para 0 y para 1, la clasificación será el de mayor resultado, para el ejemplo su salida es 1, pues quien tiene un mayor valor.

3.4.1 Algoritmo.

Aquí está el algoritmo un poco más generalizado. Para R_j de 1, 1, contamos, de nuestro conjunto de datos, cuantos ejemplos para el atributo j han sido 1 cuando el atributo también tiene valor 1 y dividimos esta cantidad por la cantidad total de ejemplos positivo o con salida 1 de nuestro conjunto de datos.

Estimación de datos para toda j cuando R es 1, 1

Estimación de datos para toda j cuando R es 0, 1

De la misma manera de 1, 0 es el numero de de ejemplos en el cual el atributo j tiene valor 1 cuando la salida es 0, divididos entre el total de ejemplos con salida negativa 0 y es 1 menos de 1, 0

Estimación de datos para toda j cuando R es 1, 0

Estimación de datos para toda j cuando R es 0, 0

Dado un nuevo ejemplo, x , el resultado cuando la salida es 1 $S(1)$, es el producto, de toda j de 1, 1 si $x_j = 1$ and de 0, 1 si $x_j = 0$

$S(1) =$

Similarmente $S(0)$, es el producto de todo j , de 1, 0 si $x_j = 1$ y de 0, 0 si $x_j = 0$

$S(0) =$

Finalmente, si $S(1) > S(0)$ entonces predeciremos que $y = 1$ de ser lo contrario $y = 0$ para el Nuevo caso.

3.4.2 Naive Bayes y Corrección de Laplace.

En nuestra hipótesis del ejercicio anterior que la salida fue 1, veremos una ligera modificación de la técnica de Naive Bayes, el cual se considera útil por dos razones la primera para que ninguna fracción sea haga 0 o 1 y la segunda porque es bastante útil cuando se tienen pocos casos.

Para ello consideramos lo siguiente para evitar un 0 o un 1 como respuesta

Estimación de datos para toda j cuando R es 1, 1

Estimación de datos para toda j cuando R es 0, 1

Estimación de datos para toda j cuando R es 1, 0

Estimación de datos para toda j cuando R es 0, 0

Para el conjunto de datos del ejemplo anterior la hipótesis resultante aplicando la corrección de Laplace es la siguiente:

F ₁	F ₂	F ₃	F ₄	Y	
0	1	1	0	1	$R_1(1,1)=2/7$ $R_1(0,1)=5/7$
0	0	1	1	1	$R_1(1,0)=6/7$ $R_1(0,0)=1/7$
1	0	1	0	1	$R_2(1,1)=2/7$ $R_2(0,1)=5/7$
0	0	1	1	1	$R_2(1,0)=3/7$ $R_2(0,0)=4/7$
0	0	0	0	1	$R_3(1,1)=5/7$ $R_3(0,1)=2/7$
1	0	0	1	0	$R_3(1,0)=2/7$ $R_3(0,0)=5/7$
1	1	0	1	0	$R_4(1,1)=3/7$ $R_4(0,1)=4/7$
1	0	0	0	0	$R_4(1,0)=5/7$ $R_4(0,0)=2/7$
1	1	0	1	0	
1	0	1	1	0	

Tabla 3.10 Cálculo de fracciones probabilísticas

Vemos que en nuestra hipótesis que la fracción del atributo 1 cuando es 0 y salida 0 ya no es 0 ahora es 1/5 el cual es más conveniente para nuestra predicción.

Evaluando el nuevo caso con la nueva hipótesis, vemos que el caso es de clase 1.

$$S(1) = R_1(0,1) * R_2(0,1) * R_3(1,1) * R_4(1,1)$$

$$S(1) = (5/7) * (5/7) * (5/7) * (3/7) = 0.156$$

$$S(0) = R_1(0,0) * R_2(0,0) * R_3(1,0) * R_4(1,0)$$

$$S(0) = (1/7) * (4/7) * (5/7) * (3/7) = 0.017$$

Implementación, Evaluación y Resultados.

Cap. 4

4. Implementación, Evaluación y Resultados.

En ésta sección abordaremos el proceso de implementación de los algoritmos de aprendizaje de máquina, los fundamentos teóricos se vieron en capítulos anteriores, ahora veamos más a detalle el proceso de trabajo. Por cada algoritmo hemos diseñado ejemplos de tal forma que sirvan para el lector la comprensión total del contenido, después de cada ejemplo y los diferentes pasos que se llevaron mostraremos las ventajas y desventajas por cada algoritmo encontrado de acuerdo a su implementación y evaluación

Recordemos que tratamos con 1 algoritmos y 3 técnicas principales, los algoritmos los abordaremos de acuerdo al orden que aparecen en la barra de opciones de programa, ver Anexo 1 ellos corresponden a:

1. Naive Bayes.
2. Árboles de Decisión.
3. Vecino más cercano.
4. Validación Cruzada.

El objetivo principal de las técnicas de aprendizaje de máquina (Aprendizaje Automatizado), es hacer predicciones, es decir, de acuerdo al caso de estudio y sus correspondientes datos de entradas predecir o generar para un nuevo caso de estudio generar una salida.

Haremos referencia a nuestro programa con el nombre de **IMEAM 1.0** en correspondencia a la Implementación y Evaluación de los Algoritmos de Aprendizaje de Máquina, tema central de este estudio monográfico.

IMEAM 1.0 además de las opciones de acceso a cada uno los algoritmos y técnicas, considera una opción “Datos”, ésta nos permite visualizar el conjunto de datos original y una organización para el tratamiento de los mismos, de tal modo que antes de comenzar con cada algoritmo hablaremos de los aspectos básicos de los datos como cantidad de registro suministrado y sus características.

Para la manipulación de IMEAM 1.0 hemos diseñado una guía que permita el manejo, esta guía la pueden encontrar el sección de anexos. IMEAM 1.0 tiene 5 funciones; 4 para los algoritmos mencionada anteriormente y otro llamado “Datos”, para visualizar nuestro conjunto de datos.

4. 1 Datos.

Aprender automáticamente requiere de un conjunto de datos, también llamado conjunto de entrenamiento, que requieren un procesamiento previo antes de aplicar los algoritmos. Se le conocen como casos o ejemplos almacenados, los algoritmos o técnicas permiten encontrar patrones de comportamientos, es decir, una hipótesis, la cual es generada por cada algoritmo de aprendizaje y enfrentada a nuevos casos. La idea clave es que la hipótesis construida pueda predecir un nuevo caso.

4.1.2 Suministración de nuestro conjunto de datos.

Utilizaremos para la evaluación de los algoritmos el conjunto de datos que corresponde a la base de datos de las notas de los estudiantes de la carrera de ingeniería en computación, la base de datos con este contenido fue proporcionada por la División de informática y tecnología de la información (DITI-UNI). Los aspectos básicos de la información se detallan en la siguiente tabla.

Base de Datos	Suministrado Por	Fechas Registro de	Cantidad de Registros
Notas de Estudiantes de Computación	DITI	Los registros están comprendido desde el I semestre del 2006 hasta registros del II semestre del 2009	Registro de 589 Estudiantes

Tabla 4.1 Información de nuestro conjunto de datos

Cada registro de notas por estudiante guarda las calificaciones por estudiantes, sus registros de notas corresponden a las actas., estos registros fueron proporcionados bajo la autorización del secretario general de la Universidad Nacional de Ingeniería. Los datos suministrado según la división de informática, son los más reciente y hasta el momento respaldado en la base de datos.

Por razones internas y la seguridad del manejo de información los carne de los estudiantes fueron editados, siguiendo siempre con un formato para identificar el año lectivo y el alumno correspondiente. Una vez que se ingresa al sistema la pantalla que se muestra es la siguiente

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

	Carnet	Asignatura	Nota	Modalidad
▶	2006-PPCPT	FISICA II	68	Regular
	2006-PPCPT	PROGRAMACI...	91	Regular
	2006-PPCPT	SISTEMAS DI...	65	Regular
	2006-PPCPT	TEORIA DE L...	60	Regular
	2006-PPCPT	ADMINISTRA...	90	Regular
	2006-PPCPT	ARQUITECTU...	76	Regular
	2006-PPCPT	ECONOMIA D...	72	Regular

Organización

	Carnet	Filosofía	Introducción a la Ingeniería	Geometría Analítica y Descriptiva	Inglés I	Matemáticas I	Conceptos	Lenguajes de Programación	Inglés II	R
*										

Figura 4.1 Datos

El objetivo de esta pantalla es hacer una conexión con la base de datos con los registros, en total se tienen 589 registros tal como se pueden observar en la segunda cuadrícula una vez que se haya dado clic en la opción organización. El botón Organización permite organizar los registros de información por estudiantes asociado respectivamente con sus clases, esta organización nos facilita el tratamiento y la manipulación de datos cuando se considera cualquiera de los algoritmos de aprendizaje.

4.2 Técnica de aprendizaje Naive Bayes.

4.2.1 Propósito al implementar esta técnica:

Realizar una predicción sobre la asignatura de Sistemas Operativos y Redes, es decir responder: ¿Cuál será el rendimiento académico del estudiante x en la asignatura de Sistemas Operativos y Redes?

4.2.2 Ventajas de la Técnica:

- ✓ La Hipótesis, que nos permite realizar una predicción, está basada en fracciones probabilísticas.
- ✓ La transformación de los datos es muy simple, en relación a datos numéricos.
- ✓ Trabaja bien con grandes volúmenes de Información.

4.2.3 Desventajas de la Técnica:

- ✓ Su transformación únicamente es booleana, por lo que los datos se tienen que generalizar en dos subconjuntos.
- ✓ La hipótesis a medida que crecen los atributos se vuelve más grande.
- ✓ No es posible saber que atributos son más significativos para la predicción de nuevos casos

4.2.4 Implementación de la Técnica.

Como ya se ha mencionado en apartados anteriores, la implementación se realizó en Microsoft Visual Studio y se utilizó como gestor de bases de datos Microsoft SQL server 2005.

La implementación de la técnica tiene un grado de complejidad baja, una vez que se maneja el mecanismo de trabajo de la técnica el proceso de programación es bastante simple, es importante mencionar que Visual Studio permite operaciones matemáticas básicas que facilitan el cálculo. Sin olvidar que es de importancia dominar en un nivel medio el gestor de base de datos SQL, pues es necesario tener conocimientos básicos de conexión, consultas y algunos aspectos básico de programación.

4.2.4.1 Caso de Estudio.

- ✓ La predicción se realizará para la asignatura de **Sistemas Operativos y Redes**.

4.2.4.2 Datos de Entrenamiento.

Recordemos que uno de los objetivos es la construcción de la hipótesis, el cual permitirá posteriormente hacer una predicción ante nuevos casos. Para ello es necesario seleccionar un conjunto de entrenamiento, el cual se analizará, para obtener nuestra hipótesis.

Nuestro conjunto de datos estará dado por todas las asignaturas anteriores a Sistemas Operativos y Redes. Dichas asignaturas corresponden a:

- | | |
|---|---|
| 1. Filosofía. | 12. Geometría Computacional. |
| 2. Introducción a la Ingeniería en Computación. | 13. Sociología y Ética. |
| 3. Geometría Analítica y Descriptiva. | 14. Matemáticas III. |
| 4. Inglés I. | 15. Física I. |
| 5. Matemáticas I. | 16. Algoritmización y Estructuras de Datos. |
| 6. Conceptos de Lenguajes. | 17. Matemáticas Discretas. |
| 7. Lenguajes de Programación. | 18. Cultura de Paz y Derechos Humanos. |
| 8. Inglés II. | 19. Física II. |
| 9. Redacción Técnica. | 20. Teoría de la Computación. |
| 10. Matemáticas II. | 21. Sistemas Digitales. |
| 11. Programación Orientada a Objetos. | 22. Programación Gráfica. |

Estas asignaturas la podemos ver en el flujograma actual de la carrera.

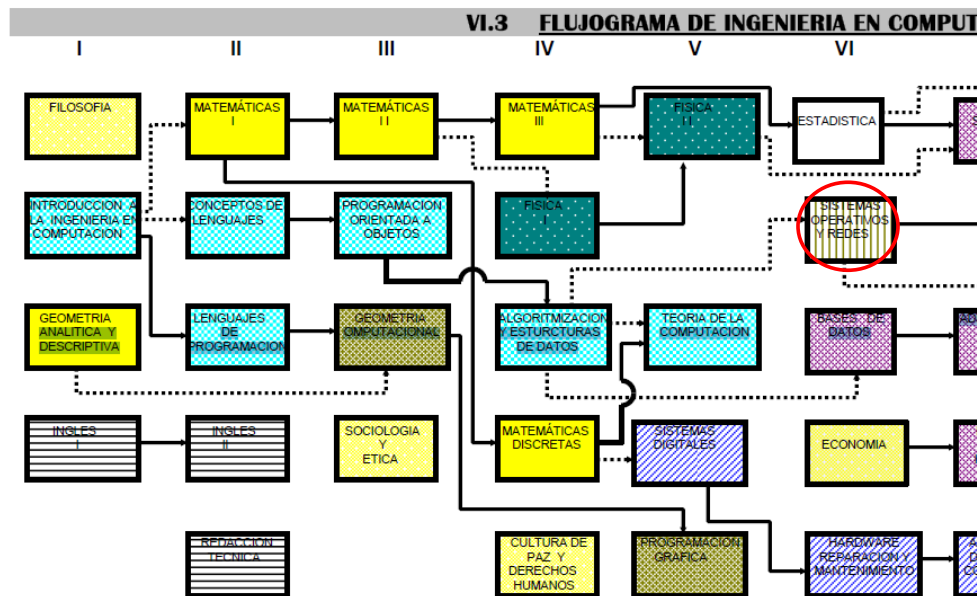


Figura 4.2 Asignaturas que serán nuestro atributo en nuestro conjunto de entrenamiento.

Las 22 asignaturas en mención corresponden al total de atributos de nuestro conjunto de entrenamiento.

4.2.4.3 Consideraciones Generales del Conjunto de Datos.

- ✓ Del conjunto total de casos hemos considerado a los estudiantes que tienen algún registro de nota, exclusivamente en la modalidad semestre Regular
- ✓ No se toman en cuenta aquellos casos en que el alumno no tiene registro en algunas de las asignaturas consideradas.
- ✓ Se consideran los registros de los estudiantes con carne 2006 y 2007, que son los estudiantes que ya han cursado las clases hasta Sistemas Operativos y Redes. Este conjunto de estudiantes será considerado para nuestro caso de estudio como los casos o ejemplos de los que la técnica de Naive Bayes aprenderá y construirá su hipótesis.

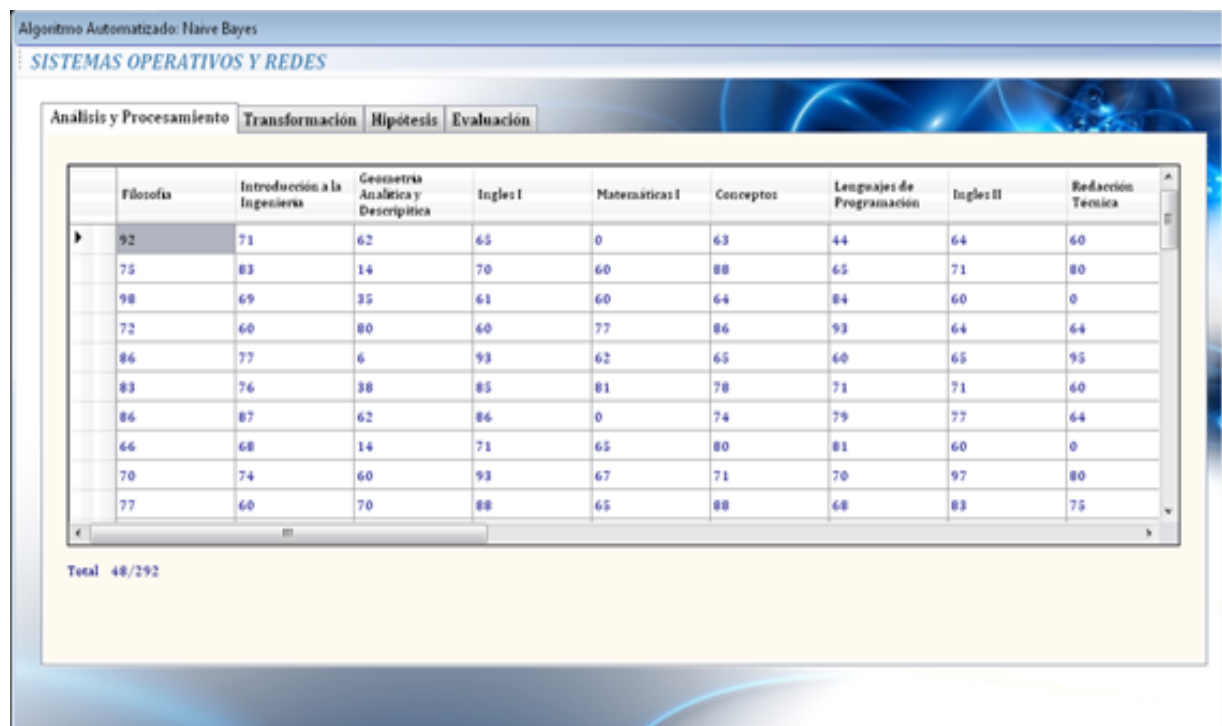
4.2.4.4 Seleccionando Nuestro Conjunto de Entrenamiento con IMEAN 1.0.

Una vez que hayamos seleccionado de nuestra barra de opciones la opción “Naive Bayes” (Ver Anexo 1). Observamos inmediatamente la selección de

nuestro conjunto de entrenamiento, tomando en cuenta las consideraciones antes expuestas.

En total se encontraron 292 estudiantes que ya debieron haber cursado la Asignatura de Sistemas Operativos y Redes, sin embargo las personas que ya han cursado¹ todas las asignaturas anteriores e incluyendo Sistemas Operativos y Redes corresponden a un subtotal de 48 estudiantes de los 292.

De estos 48 casos, formaremos nuestra hipótesis aplicando la técnica de Naive Bayes. La interfaz grafica correspondiente a nuestro conjunto de entrenamiento es mostrada a continuación:



The screenshot shows a software interface titled 'Algoritmo Automatizado: Naive Bayes'. Below the title is a tabbed menu with four options: 'Análisis y Procesamiento' (selected), 'Transformación', 'Hipótesis', and 'Evaluación'. The main area displays a table with 9 columns: 'Filosofía', 'Introducción a la Ingeniería', 'Geometría Analítica y Descriptiva', 'Inglés I', 'Matemáticas I', 'Conceptos', 'Lenguajes de Programación', 'Inglés II', and 'Redacción Técnica'. The table contains 12 rows of student data. The first row is highlighted. At the bottom left of the table area, it says 'Total 48/292'.

	Filosofía	Introducción a la Ingeniería	Geometría Analítica y Descriptiva	Inglés I	Matemáticas I	Conceptos	Lenguajes de Programación	Inglés II	Redacción Técnica
▶	92	71	62	65	0	63	44	64	60
	75	83	14	70	60	88	65	71	80
	98	69	35	61	60	64	84	60	0
	72	60	80	60	77	86	93	64	64
	86	77	6	93	62	65	60	65	95
	83	76	38	85	81	78	71	71	60
	86	87	62	86	0	74	79	77	64
	66	68	14	71	65	80	81	60	0
	70	74	60	93	67	71	70	97	80
	77	60	70	88	65	88	68	83	75

Total 48/292

Figura 4.3 Casos a considerar la construcción de hipótesis

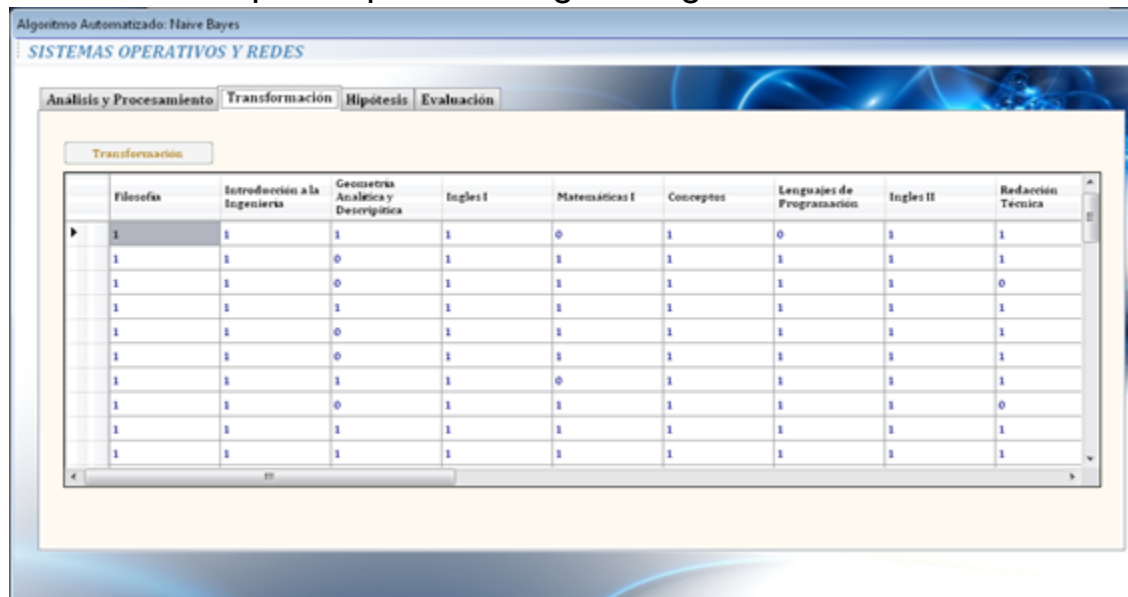
4.2.4.5 Transformación de Datos con IMEAM 1.0

La Técnica de Naive Bayes requiere de una transformación de los datos originales a datos booleanos, ésta transformación es necesaria pues de aquí es que se parte para establecer nuestras fracciones probabilísticas. La transformación de todo nuestro conjunto de datos se realiza tomando en cuenta las siguientes condiciones:

- ✓ Si la Nota es menor a 60 su transformación corresponderá a un 0.
- ✓ Si la Nota es mayor o igual a 60 su transformación corresponderá a un 1.

¹ La nota puede estar aprobada o reprobada.

Esta transformación del conjunto de datos (conjunto de entrenamiento) seleccionados se puede apreciar en la gráfica siguiente:



	Filosofía	Introducción a la Ingeniería	Geometría Analítica y Descriptiva	Inglés I	Matemáticas I	Conceptos	Lenguajes de Programación	Inglés II	Redacción Técnica
1	1	1	1	1	0	1	0	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1
1	1	1	0	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Figura 4.4 Transformación de nuestro conjunto de datos

Paso 1.

Observemos que todos los campos han sido transformados a *ceros* y *unos* incluyendo la salida que es la asignatura de Sistemas Operativos y Redes.

Es importante mencionar que de igual forma al evaluar un nuevo caso, su salida también corresponderá a un 0 o a un 1, en caso de que sea un cero, sabemos que por las condiciones de su transformación la salida corresponderá a una nota menor que 60 y en caso contrario a una nota mayor o igual a 60.

El mecanismo básico de trabajo de la técnica de Naive Bayes, es evaluar cada atributo, en nuestro caso 22 atributos, y encontrar 4 razones:

1. La fracción de todos los ejemplos positivos cuando el atributo en cuestión es 1. **R(1,1).**
2. La fracción de todos los ejemplos positivos cuando el atributo en cuestión es 0. **R(0,1).**
3. La fracción de todos los ejemplos negativos cuando el atributo en cuestión es 1. **R(1,0).**
4. La fracción de todos los ejemplos negativos cuando el atributo en cuestión es 0. **R(0,0).**

Considerando esto, para nuestro caso de estudio, en total se ha calculado un total de 88 fracciones, este conjunto de fracciones resultante es nuestra hipótesis. A continuación se muestra nuestra hipótesis resultante.

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	47/47	0/47	1/1	0/1
2	47/47	0/47	1/1	0/1
3	28/47	19/47	0/1	1/1
4	47/47	0/47	1/1	0/1
5	35/47	12/47	0/1	1/1
6	47/47	0/47	1/1	0/1
7	45/47	2/47	1/1	0/1
8	45/47	2/47	0/1	1/1
9	38/47	9/47	1/1	0/1
10	45/47	2/47	1/1	0/1
11	46/47	1/47	1/1	0/1
12	47/47	0/47	1/1	0/1
13	39/47	8/47	1/1	0/1
14	44/47	3/47	1/1	0/1
15	32/47	15/47	1/1	0/1
16	46/47	1/47	1/1	0/1
17	46/47	1/47	1/1	0/1
18	46/47	1/47	1/1	0/1
19	39/47	8/47	1/1	0/1
20	47/47	0/47	1/1	0/1
21	45/47	2/47	1/1	0/1
22	47/47	0/47	1/1	0/1

Figura 4.5 Hipótesis

El conjunto de fracciones mostradas en la gráfica anterior corresponde a nuestra hipótesis que permitirá generar la salida para nuevos casos.

Paso 2.

El siguiente paso, es hacer dos cálculos uno en donde se considera las fracciones de los ejemplos positivos y el otros es considerando las fracciones de los ejemplos negativos.

Consideremos que tenemos el siguiente caso,

Asignatura	Nota
Filosofía.	65
Introducción a la Ingeniería en Computación.	88
Geometría Analítica y Descriptiva.	100
Ingles I.	75
Matemáticas I.	100
Conceptos de Lenguajes.	0
Lenguajes de Programación.	74
Ingles II.	48
Redacción Técnica.	49
Matemáticas II.	50
Programación Orientada a Objetos.	100
Geometría Computacional.	87
Sociología y Ética.	0
Matemáticas III.	90
Física I.	10
Algoritmización y Estructuras de Datos.	100
Matemáticas Discretas.	50
Cultura de Paz y Derechos Humanos.	50
Física II.	60
Teoría de la Computación.	67
Sistemas Digitales.	80
Programación Gráfica.	100

Tabla 4.2 Nuevo caso

Lo que se quiere, dado estas notas, es predecir que si el estudiante tendrá una nota mayor o igual a 60 o menor a 60. Para ello introducimos los datos en nuestra interfaz.

Paso 3.

Luego que se introduce el nuevo caso, el programa procede a realizar la transformación correspondiente al nuevo caso. Las siguientes interfaces muestran donde introducir los datos.

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

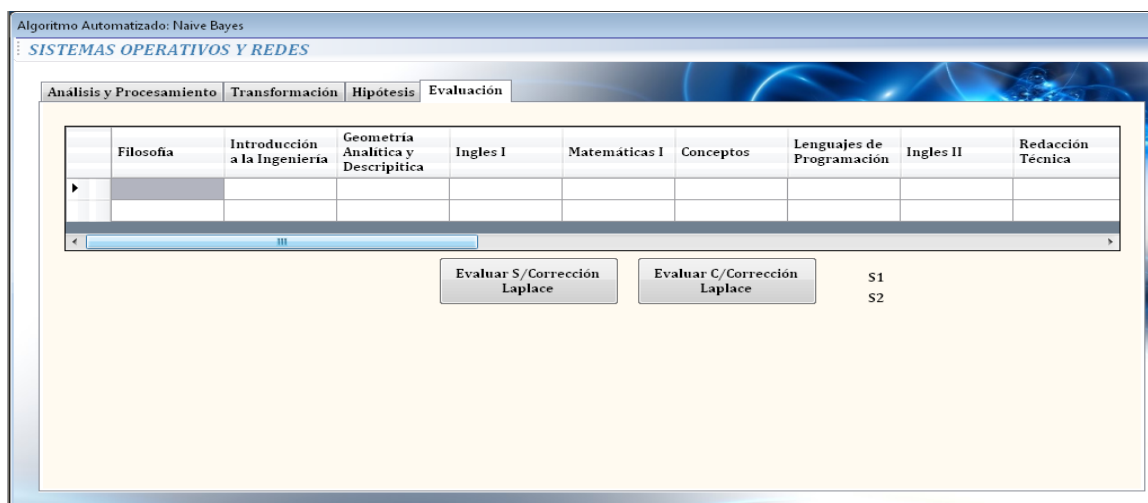


Figura 4.6 Interfaz para evaluación del nuevo caso.

Después de introducir el nuevo caso y al seleccionar la Opción “Evaluar S/Corrección de Laplace” .Automáticamente hace la transformación de los datos y genera la salida correspondiente. La transformación se muestra a continuación:

Asignatura	Nota	Transformación
Filosofía.	65	1
Introducción a la Ingeniería en Computación.	88	1
Geometría Analítica y Descriptiva.	100	1
Inglés I.	75	1
Matemáticas I.	100	1
Conceptos de Lenguajes.	0	0
Lenguajes de Programación.	74	1
Inglés II.	48	0
Redacción Técnica.	49	0
Matemáticas II.	50	0
Programación Orientada a Objetos.	100	1
Geometría Computacional.	87	1
Sociología y Ética.	0	0
Matemáticas III.	90	1
Física I.	10	0
Algoritmización y Estructuras de Datos.	100	1
Matemáticas Discretas.	50	0
Cultura de Paz y Derechos Humanos.	50	0
Física II.	60	1
Teoría de la Computación.	67	1
Sistemas Digitales.	80	1
Programación Gráfica.	100	1

Tabla 4.3 Transformación del nuevo caso

Para encontrar el valor de $S(1)$ y $S(0)$, las multiplicaciones de las fracciones será la siguiente:

Atributo	Valor	$S(1)$	$S(0)$
		Fracción	Fracción
1	1	$R_1(1,1)$	$R_1(1,0)$
2	1	$R_2(1,1)$	$R_2(1,0)$
3	1	$R_3(1,1)$	$R_3(1,0)$
4	1	$R_4(1,1)$	$R_4(1,0)$
5	1	$R_5(1,1)$	$R_5(1,0)$
6	0	$R_6(0,1)$	$R_6(0,0)$
7	1	$R_7(1,1)$	$R_7(1,0)$
8	0	$R_8(0,1)$	$R_8(0,0)$
9	0	$R_9(0,1)$	$R_9(0,0)$
10	0	$R_{10}(0,1)$	$R_{10}(0,0)$
11	1	$R_{11}(1,1)$	$R_{11}(1,0)$
12	1	$R_{12}(1,1)$	$R_{12}(1,0)$
13	0	$R_{13}(0,1)$	$R_{13}(0,0)$
14	1	$R_{14}(1,1)$	$R_{14}(1,0)$
15	0	$R_{15}(0,1)$	$R_{15}(0,0)$
16	1	$R_{16}(1,1)$	$R_{16}(1,0)$
17	0	$R_{17}(0,1)$	$R_{17}(0,0)$
18	0	$R_{18}(0,1)$	$R_{18}(0,0)$
19	1	$R_{19}(1,1)$	$R_{19}(1,0)$
20	1	$R_{20}(1,1)$	$R_{20}(1,0)$
21	1	$R_{21}(1,1)$	$R_{21}(1,0)$
22	1	$R_{22}(1,1)$	$R_{22}(1,0)$

Tabla 4.4 Evaluación del nuevo caso.

Paso 4.

La multiplicación lo realiza automáticamente el sistema, cuyo resultados son los siguientes:

- ✓ Para $S(1)=0$
- ✓ Para $S(0)=0$

La salida corresponde al de mayor resultado entre ambos, puesto que en este caso ambas salidas son iguales, podemos dar 2 opciones de respuesta:

1. Tomar arbitrariamente cualquier valor y predecir con respecto a ella.
2. Decir que la probabilidad de que el estudiante, tenga una nota mayor o igual a 60 es del 50%.

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

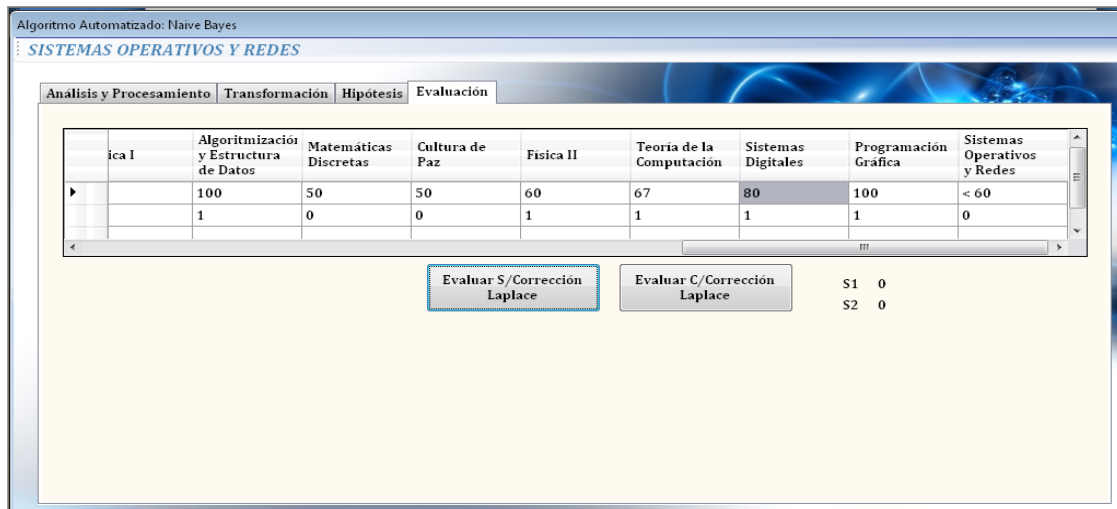


Figura 4.7 Interfaz de evaluación del nuevo caso.

Como podemos ver, aquí se muestra una desventaja de la técnica, sin embargo ésta técnica se auxilia de un método llamado **Corrección de Laplace**, abordado en el capítulo 3 sección 3.4.2 Este método lo que hace es evitar resultados absolutos que sean 0 ó 1, para evitar salidas iguales como el ejemplo anterior. El método de la Corrección de Laplace consiste en sumar un 1 al numerador y un 2 al denominador de cada fracción. Como se muestra en la siguiente figura:

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	48/49	1/49	2/3	1/3
2	48/49	1/49	2/3	1/3
3	29/49	20/49	1/3	2/3
4	48/49	1/49	2/3	1/3
5	36/49	13/49	1/3	2/3
6	48/49	1/49	2/3	1/3
7	46/49	3/49	2/3	1/3
8	46/49	3/49	1/3	2/3
9	39/49	10/49	2/3	1/3
10	46/49	3/49	2/3	1/3
11	47/49	2/49	2/3	1/3
12	48/49	1/49	2/3	1/3
13	40/49	9/49	2/3	1/3
14	45/49	4/49	2/3	1/3
15	33/49	16/49	2/3	1/3
16	47/49	2/49	2/3	1/3
17	47/49	2/49	2/3	1/3
18	47/49	2/49	2/3	1/3
19	40/49	9/49	2/3	1/3
20	48/49	1/49	2/3	1/3
21	46/49	3/49	2/3	1/3
22	48/49	1/49	2/3	1/3

Figura 4.8 Hipótesis

Considerando ahora nuestra nueva hipótesis al realizar el cálculo para $S(1)$ y para $S(0)$, se tiene:

$$S(1)=0.000000000364320021000008$$

$$S(0)=0.0000000261049183873019$$

Por tanto ahora con claridad podemos predecir que la salida es clase 0 (porque $S(0)$ es mayor que $S(1)$) y de acuerdo a nuestra transformación, esto indica que se obtendrá una nota menor a 60. El estudiante se aplazara en la asignatura de Sistemas Operativos y Redes.

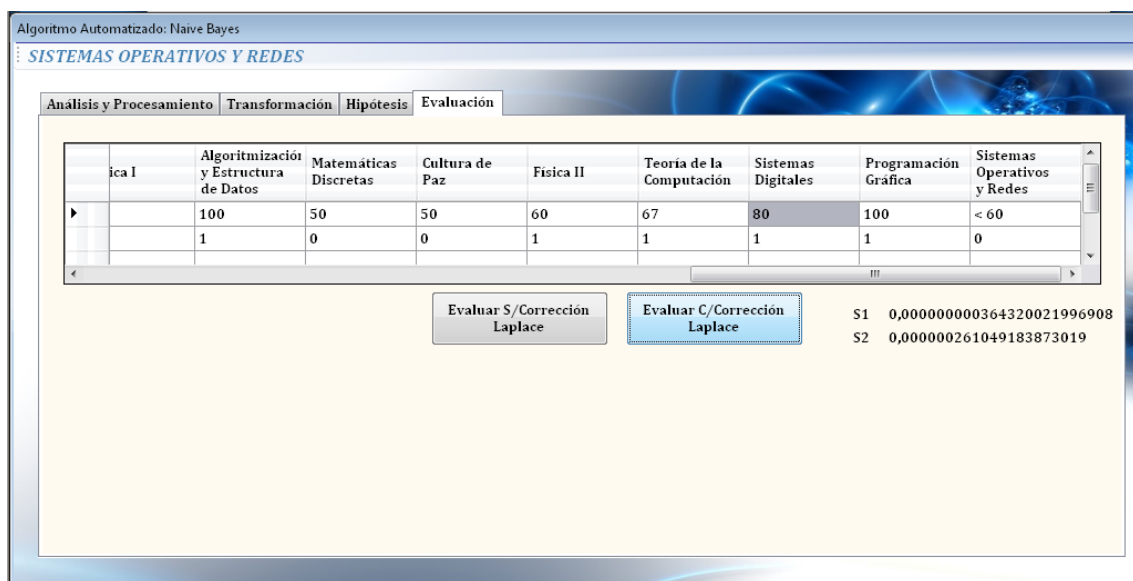


Figura 4.9 Interfaz de evaluación del nuevo caso con corrección de Laplace

4.3 Algoritmo de Aprendizaje: Árboles de Decisión.

4.3.1 Propósito al implementar esta técnica:

Realizar una predicción sobre la asignatura de Estadística, para ello se toma en cuenta las notas de todas las asignaturas anteriores a dicha clase.

¿Qué asignatura inciden para aprobar la asignatura de Estadística?

4.3.2 Ventajas de la Técnica:

- ✓ La hipótesis, representada a través de un árbol de decisión, permite ver los atributos que más inciden en la predicción de un caso.
- ✓ Facilidad para interpretar la hipótesis, debido a que es posible su representación gráfica

4.3.3 Desventajas:

- ✓ El árbol de decisiones para datos con muchos atributos puede tener un tamaño considerable.
- ✓ La construcción del árbol es dinámica y se requiere de gran dominio en programación dinámica.

4.3.4 Implementación del Algoritmo.

Al igual que en la técnica anterior, se realizó en Microsoft Visual Studio y se utilizó como gestor de base de datos Microsoft SQL server 2005.

La implementación del algoritmo tiene una complejidad media, dado el comportamiento del algoritmo basado en cálculo de entropía de cada atributo más de una vez.

4.3.4.1 Caso Estudio.

- ✓ La construcción de la hipótesis en forma de árbol que permitirá predecir si un estudiante x aprobará la asignatura de **Estadísticas**.

Nuestro conjunto de datos estará dado por todas las asignaturas anteriores a Estadística. Dichas asignaturas corresponden a:

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

- | | |
|---|--|
| <ol style="list-style-type: none"> 1. Filosofía. 2. Introducción a la Ingeniería en Computación. 3. Geometría Analítica y Descriptiva. 4. Ingles I. 5. Matemáticas I. 6. Conceptos de Lenguajes. 7. Lenguajes de Programación. 8. Ingles II. 9. Redacción Técnica. 10. Matemáticas II. 11. Programación Orientada a Objetos. | <ol style="list-style-type: none"> 12. Geometría Computacional. 13. Sociología y Ética. 14. Matemáticas III. 15. Física I. 16. Algoritmización y Estructuras de Datos. 17. Matemáticas Discretas. 18. Cultura de Paz y Derechos Humanos. 19. Física II. 20. Teoría de la Computación. 21. Sistemas Digitales. 22. Programación Grafica. |
|---|--|

Estas asignaturas la podemos ver en el flujograma actual de la carrera.

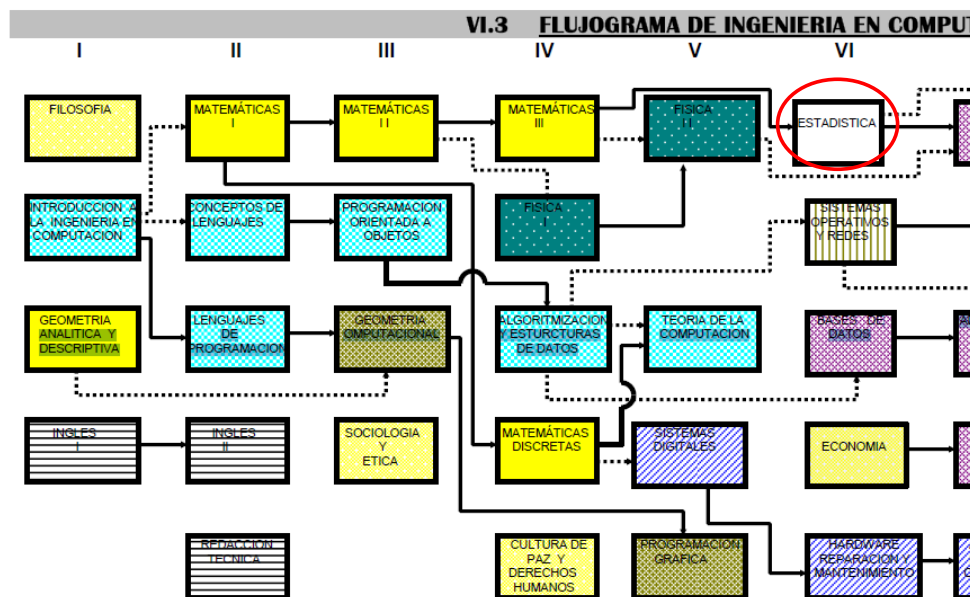


Figura 4.10 Atributos de nuestro conjunto de entrenamiento.

Las 22 asignaturas en mención corresponden al total de atributos de nuestro conjunto de entrenamiento.

4.3.4.2 Consideraciones Generales del Conjunto de Datos.

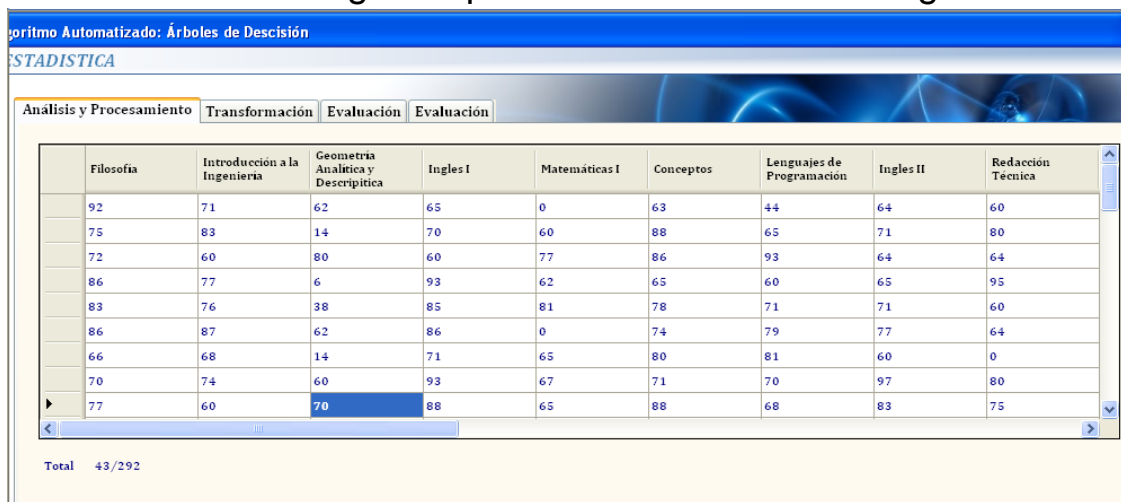
- ✓ Del conjunto total de casos hemos considerado a los estudiantes que tienen algún registro de nota, exclusivamente en la modalidad semestre Regular
- ✓ No se toman en cuenta aquellos casos en que el alumno no tiene registro en algunas de las asignaturas consideradas.
- ✓ Se considera los registros de los estudiantes con carne 2006 y carne 2007. El conjunto de estudiantes que ya han cursado las asignaturas hasta Estadística, este será considerado para nuestro caso de estudio. Son los casos o ejemplos de los que el algoritmo de Árbol de Decisión aprenderá y construirá su hipótesis

4.3.4.3 Seleccionando Nuestro Conjunto de Entrenamiento IMEAM 1.0.

Una vez que hayamos seleccionado de nuestra barra de opciones la opción “Árbol de Decisión” (Ver Anexo 1). Observamos inmediatamente la selección de nuestro conjunto de entrenamiento, tomando en cuenta las consideraciones antes expuestas.

En total se encontraron 292 estudiantes que ya debieron haber cursado la Asignatura de Estadística, sin embargo las personas que ya han cursado² todas las asignaturas anteriores e incluyendo Estadística corresponden a un subtotal de 43 estudiantes de los 292.

De estos 43 casos, formaremos nuestra hipótesis aplicando el algoritmo de Árbol de Decisión. La interfaz gráfica que muestra estos casos es la siguiente:



Algoritmo Automatizado: Árboles de Decisión

ESTADISTICA

Análisis y Procesamiento Transformación Evaluación Evaluación

	Filosofía	Introducción a la Ingeniería	Geometría Analítica y Descriptiva	Inglés I	Matemáticas I	Conceptos	Lenguajes de Programación	Inglés II	Redacción Técnica
	92	71	62	65	0	63	44	64	60
	75	83	14	70	60	88	65	71	80
	72	60	80	60	77	86	93	64	64
	86	77	6	93	62	65	60	65	95
	83	76	38	85	81	78	71	71	60
	86	87	62	86	0	74	79	77	64
	66	68	14	71	65	80	81	60	0
	70	74	60	93	67	71	70	97	80
	77	60	70	88	65	88	68	83	75

Total 43/292

² La nota puede estar aprobada o reprobada.

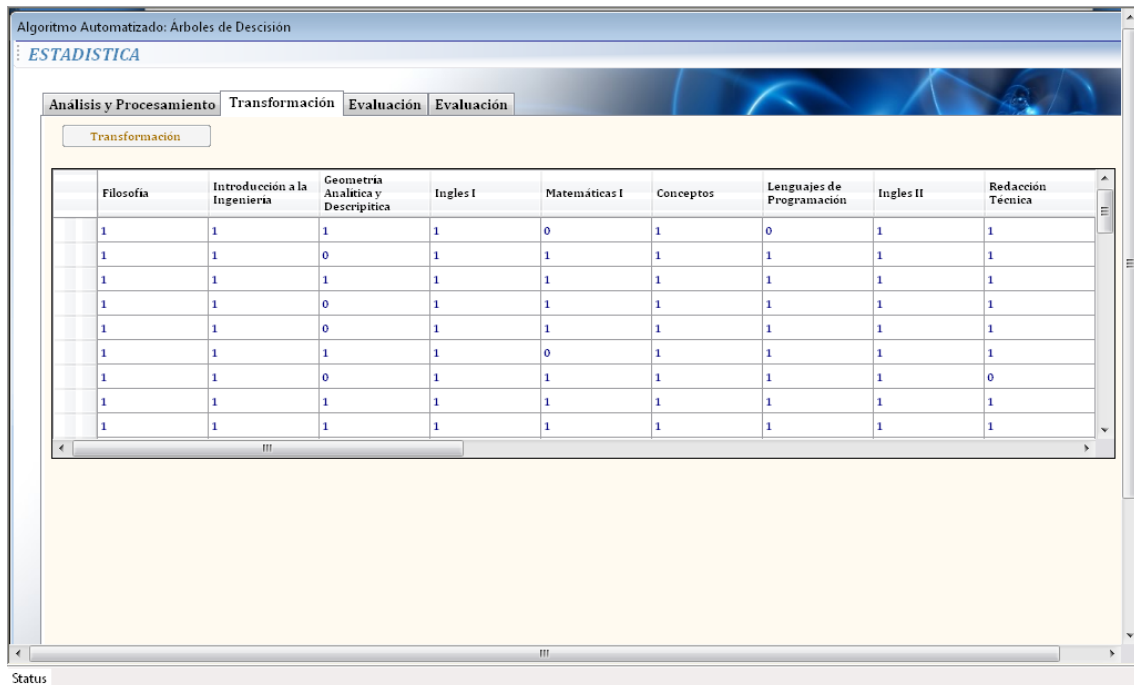
Figura 4.11 Selección de casos para construcción de hipótesis

4.3.4.4 Transformación de Datos con IMEAM 1.0

Para continuar, el algoritmo Árboles de Decisión requiere que sus datos se transformen a datos booleanos, ésta transformación es necesaria para evaluar el algoritmo. Para realizar la transformación de todo nuestro conjunto de datos se han tomado en cuenta las siguientes condiciones:

- ✓ Si la Nota es menor a 60 su transformación corresponderá a un 0.
- ✓ Si la Nota es mayor o igual a 60 su transformación corresponderá a un 1.

Esta transformación del conjunto de datos (conjunto de entrenamiento) seleccionados se puede apreciar en la gráfica siguiente:



Algorithm Automatized: Árboles de Decisión

ESTADISTICA

Analisis y Procesamiento Transformación Evaluación Evaluación

Transformación

	Filosofía	Introducción a la Ingeniería	Geometría Analítica y Descriptiva	Inglés I	Matemáticas I	Conceptos	Lenguajes de Programación	Inglés II	Redacción Técnica
1	1	1	1	1	0	1	0	1	1
1	1	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1
1	1	0	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Status

Figura 4.12 Interfaz de transformación de casos

Observemos que todos los campos han sido transformados a *ceros* y *unos* incluyendo la salida que es la asignatura de Estadística. Es importante mencionar que de igual forma al evaluar un nuevo caso, su salida también corresponderá a un 0 o a un 1, en caso de que sea un cero, sabemos que por las condiciones de su transformación la salida corresponderá a una nota menor que 60 y en caso contrario a una nota mayor o igual a 60.

Recordemos que el mecanismo básico de trabajo del Algoritmo Árboles de Decisión, es evaluar cada atributo y calcular una entropía promedio por cada uno de nuestros 22 atributos, hacer nuestro el de menor entropía y luego separar los casos en el que el atributo seleccionado es 0 y los casos en el que los atributos seleccionados sean 1.

4.3.4.5 Entropía

Los cálculos correspondientes a la entropía promedio por cada atributo se muestran a continuación:

1. Filosofía.	0,410181821478836
2. Introducción a la Ingeniería en Computación.	0,410181821478836
3. Geometría Analítica y Descriptiva.	0,804247355401948
4. Ingles I.	0,410181821478836
5. Matemáticas I.	0,609459473168319
6. Conceptos de Lenguajes.	0,410181821478836
7. Lenguajes de Programación.	0,361980316004193
8. Ingles II.	0,538000293891361
9. Redacción Técnica.	0,634864077912185
10. Matemáticas II.	0,386721521021329
11. Programación Orientada a Objetos.	0,386721521021329
12. Geometría Computacional.	0,410181821478836
13. Sociología y Ética.	0,6466673054513
14. Matemáticas III.	0,410181821478836
15. Física I.	0,553520522520712
16. Algoritmización y Estructuras de Datos.	0,485603447029056
17. Matemáticas Discretas.	0,485603447029056
18. Cultura de Paz y Derechos Humanos.	0,485603447029056
19. Física II.	0,468251557853453
20. Teoría de la Computación.	0,410181821478836
21. Sistemas Digitales.	0,538000293891361
22. Programación Gráfica.	0,410181821478836

Tabla 4.5 Cálculo de Entropías

Como podemos observar en la grafica, el atributo que tiene menor entropía promedio corresponde a la asignatura de Física I. Por tanto seleccionamos este primer nodo siendo nuestro nodo raíz de nuestro árbol, luego separamos los casos en el que la asignatura tiene valor 0 y los casos en el que tiene valor 1.

El siguiente paso será, calcular la entropía promedio por cada subconjunto y de esta forma ir construyendo nuestro árbol hasta que nuestros nodos hojas tengan 0 ó 1. Quedando nuestro árbol resultante de la siguiente manera mostrado en la figura:

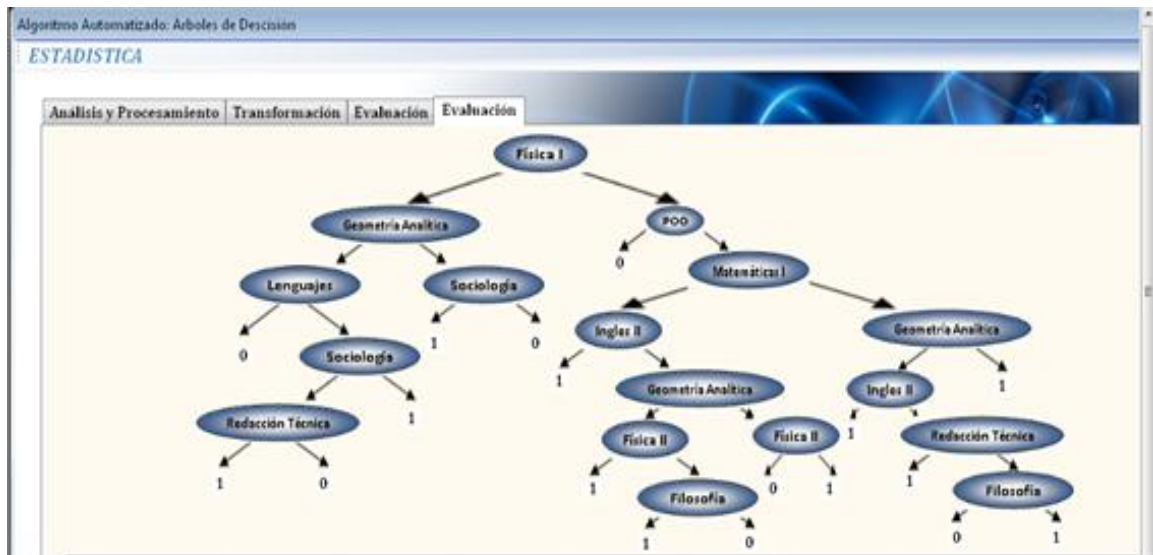


Figura 4.13 Árbol generado

Cuando tenemos el árbol generado por el algoritmo, los nuevos casos son enfrentados tomando como referencia el árbol que corresponde a nuestra hipótesis. Vemos que a diferencia de la técnica de Naive Bayes, el nuevo caso está sujeto a las asignaturas representadas en los nodos de nuestro árbol y no al total de asignaturas.

Considerando el ingreso de este nuevo caso: **87 100 0 65 78 87 60 50 58 100** y tomando en cuenta las clases que solo están representadas en el árbol. La salida generada para este caso es de clase **1** y de acuerdo a nuestra transformación **1** indica una nota mayor o igual que **60**; lo que hacemos es recorrer el árbol de acuerdo al valor transformado por cada asignatura.

Filosofía	Geometría Analítica y Descriptiva.	Matemáticas I.	Lenguaje de Programación.	Inglés II.	Redacción Técnica.	P.O. O.	Sociología y Ética.	Física I.	Física II.
87	100	0	65	78	87	60	50	58	100
1	1	0	1	1	1	1	0	0	1

Tabla 4.6 Nuevo caso a evaluar

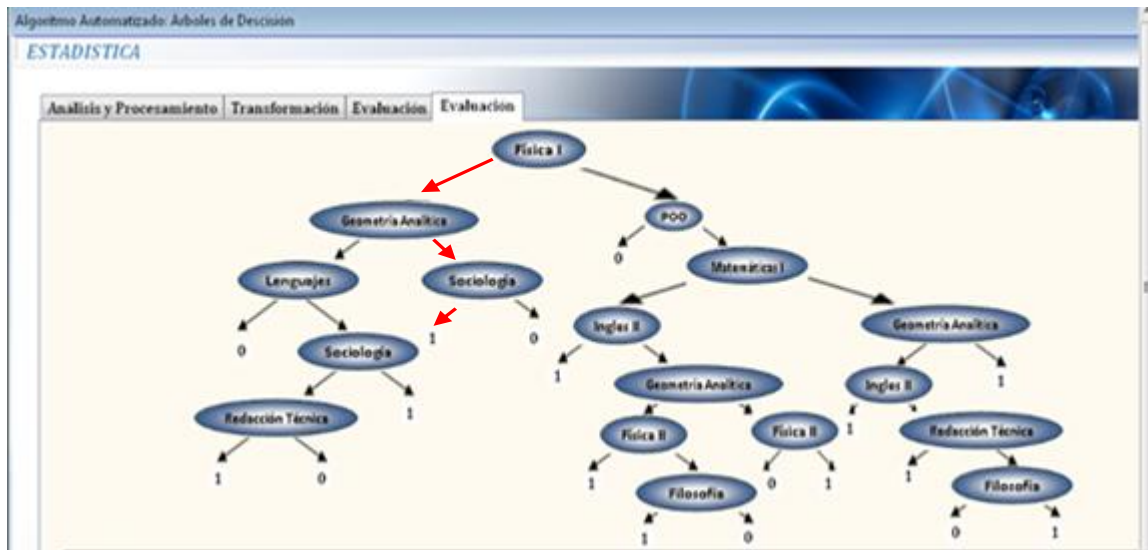


Figura 4.14 Representación de la evaluación del nuevo caso

Observamos que de acuerdo a las asignaturas y su transformación booleana, el recorrido del árbol da una salida 1

Por tanto concluimos que para este caso el estudiante aprobará la clase de Estadística la primera vez que matricule dicha asignatura.

4.4 Técnica de Aprendizaje: Vecino más Cercano.

4.4.1 Propósito al implementar esta técnica:

Realizar una predicción sobre la clase de Programación Gráfica, para ello se toma en cuenta las notas de las asignaturas de prerequisites de dicha clase.

¿Puede un estudiante x aprobar la asignatura Programación Gráfica?

4.4.2 Ventajas de la Técnica:

- ✓ La transformación no es únicamente booleana.
- ✓ No se requiere transformar datos.
- ✓ Genera una respuesta no booleana

4.4.3 Desventajas:

- ✓ El inconveniente de este método es que no realiza ninguna abstracción del problema. Esto es, la hipótesis calculada no es más sencilla que los datos de entrenamiento.
- ✓ No existe hipótesis explícita dado su mecanismo de trabajo.

4.4.4 Implementación del Algoritmo.

La técnica de vecino más cercano, se implemento en Microsoft Visual Studio y se utilizó como gestor de base de datos Microsoft SQL server 2005.

La implementación del algoritmo tiene una complejidad media, dado que el objetivo es asociar la salida del nuevo caso con los datos más cercanos de nuestro conjunto de entrenamiento. Por tanto se requiere de conocimiento de programación básica, estructuras de control, comparación de valores, etc.

4.4.4.1 Caso Estudio.

- ✓ Se realizará las predicciones en relación a la asignatura de **Programación Gráfica**, considerando como atributos las asignaturas de precedencia.

Nuestro conjunto de datos estará dado por todas las asignaturas que son prerequisite de la asignatura de Programación Gráfica. Dichas asignaturas corresponden a:

1. Introducción a la Ingeniería en Computación.
2. Lenguajes de Programación.
3. Geometría Computacional.

Estas asignaturas la podemos ver indicadas en el flujograma actual de la carrera

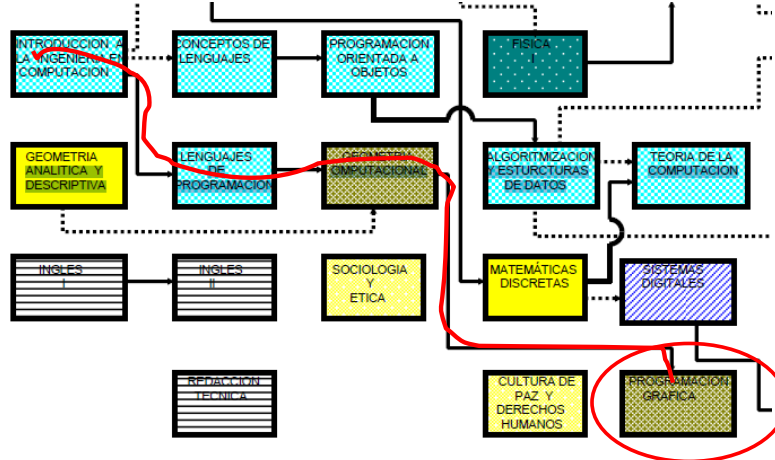


Figura 4.15 Selección de nuestros atributos.

Las 3 asignaturas en mención corresponden al total de atributos de nuestro conjunto de entrenamiento, como ya se había mencionado

4.4.4.2 Consideraciones Generales del Conjunto de Datos.

- ✓ Del conjunto total de casos hemos considerado a los estudiantes que tienen algún registro de nota, exclusivamente en la modalidad semestre Regular
- ✓ Se considera los registros de los estudiantes con carne 2006, 2007 y carne 2008. El conjunto de estudiantes que ya han cursado las clases hasta Programación Gráfica, será considerado para nuestro caso de estudio como los casos o ejemplos de los que la técnica de vecinos más cercano tomará en cuenta para predecir la salida de nuevos casos.

4.4.4.3 Seleccionando Nuestro Conjunto de Entrenamiento con IMEAM 1.0.

Una vez que hayamos seleccionado de nuestra barra de opciones la opción “Vecinos más Cercano” (Ver Anexo 1). Observamos inmediatamente la selección de nuestro conjunto de entrenamiento, tomando en cuenta las consideraciones antes expuestas.

En total se encontraron 433 estudiantes que ya debieron haber cursado la Asignatura de Programación Gráfica, sin embargo las personas que ya han cursado³ todas las asignaturas anteriores e incluyendo Programación Gráfica corresponden a un subtotal de 155 estudiantes de los 433.

De estos 155 casos, aplicaremos la técnica de vecino más cercano. La interfaz gráfica que muestra estos casos es la siguiente:

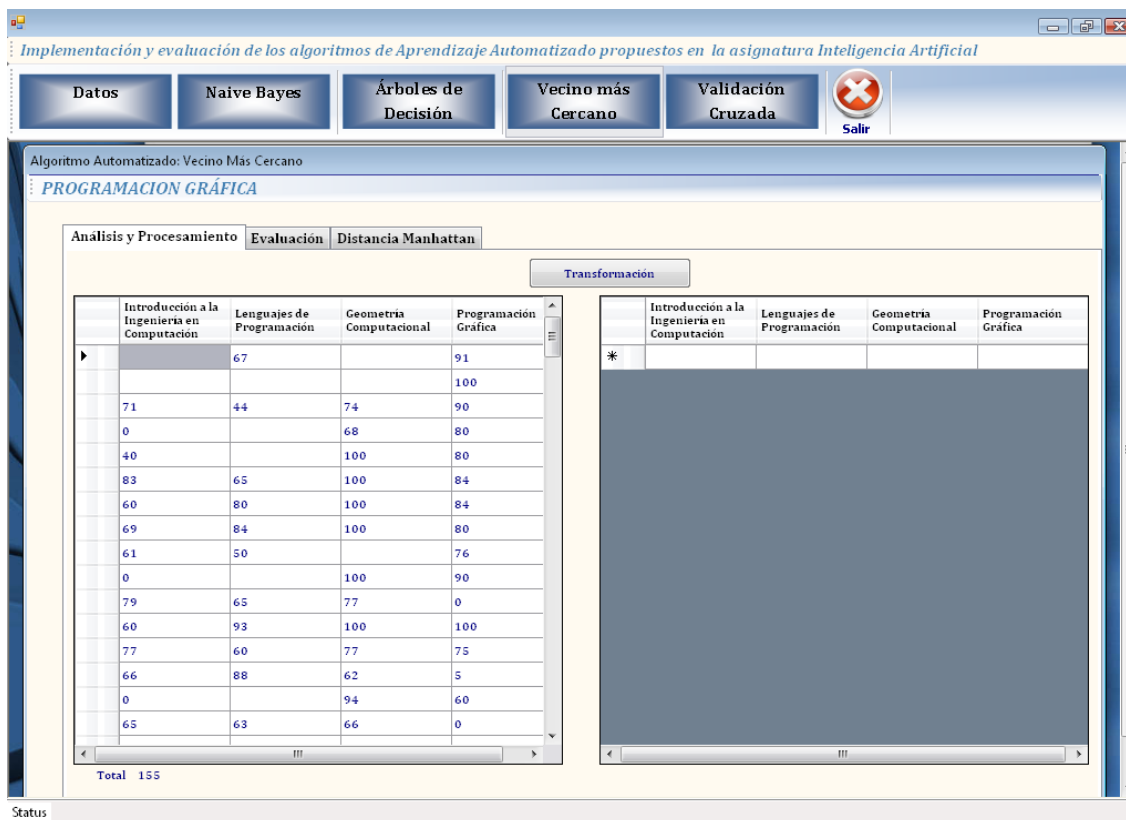


Figura 4.16 Interfaz de selección de nuestros casos.

³ La nota puede estar aprobada o reprobada.

4.4.4.4 Transformación de Datos con IMEAM 1.0

A diferencia de los algoritmos anteriores, también se tomara en cuenta aquellos casos que aunque no tengan valores en sus atributos si lo tengan en su salida correspondiente a Programación Grafica. Para lo cual requiere que se transformen sus datos, ésta transformación es necesaria para evaluar posteriormente los nuevos casos. Para realizar la transformación de todo nuestro conjunto de datos se han tomado en cuenta las siguientes condiciones:

Si la nota está entre el rango de 91 a 100 su transformación equivale a "1000"
Si la nota está entre el rango de 80 a 90 su transformación equivale a "0100"
Si la nota está entre el rango de 70 a 79 su transformación equivale a "0010"
Si la nota está entre el rango de 60 a 69 su transformación equivale a "0001"
Si la nota está entre el rango de 0 a 59 su transformación equivale a "0000"
Si el dato es nulo su transformación equivale a "1111"

Esta transformación del conjunto de datos (conjunto de entrenamiento) seleccionados se puede apreciar en la gráfica siguiente:

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
		67		91
				100
	71	44	74	90
	0		68	80
	40		100	80
	83	65	100	84
	60	80	100	84
	69	84	100	80
	61	50		76
	0		100	90
	79	65	77	0
	60	93	100	100
	77	60	77	75
	66	88	62	5
	0		94	60
	65	63	66	0
Total 155				

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
	1111	0001	1111	1000
	1111	1111	1111	1000
	0010	0000	0010	0100
	0000	1111	0001	0100
	0000	1111	1000	0100
	0100	0001	1000	0100
	0001	0100	1000	0100
	0001	0100	1000	0100
	0001	0000	1111	0010
	0000	1111	1000	0100
	0010	0001	0010	0000
	0001	1000	1000	1000
	0010	0001	0010	0010
	0001	0100	0001	0000
	0000	1111	1000	0001
	0001	0001	0001	0000

Figura 4.17 transformación de nuestros casos

Observemos que todos los campos han sido transformados de acuerdo a los rangos establecidos, incluyendo la salida que es la asignatura de Programación Gráfica. Es importante mencionar que de igual forma al evaluar un nuevo caso, su salida también corresponderá a una de las 6 transformaciones, y dependiendo de la salida corresponderá al rango de nota. Por ejemplo si la salida del nuevo caso es 0100, sabemos por los criterios de transformación de datos, que es una nota en el rango de 80 a 90

Una vez realizada la transformación hacemos la evaluación de nuevos casos, para ello ingresamos notas correspondientes a los atributos o a las asignaturas en este caso a Introducción a la Ingeniería, Lenguajes de Programación y Geometría Computacional.

Consideremos el siguiente caso a

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
90	60	87	?

Tabla 4.7 Caso a Evaluar

Es necesario saber en relación a qué cantidad de vecinos se evaluará el nuevo caso. Los casos vecinos no son más que los casos más parecidos que existen en nuestro conjunto de entrenamiento. Por ejemplo para el ejemplo a considerar primero se realiza su transformación, de acuerdo a las transformaciones.

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
90	60	87	?
0100	0001	0100	?

Tabla 4.8 Transformación de nuestro caso.

Después de la transformación buscamos los casos más parecidos en relación a la asignatura y a su valor, dentro nuestro conjunto de entrenamiento transformado.

Para el ejemplo los casos vecinos ya ordenados son:

Casos Vecinos				
Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica	Atributos Iguales
0100	0001	0100	0001	3
0100	0001	1000	0100	2
0100	0001	0010	0010	2
0100	0001	0000	0010	2
0100	0100	0100	0010	2
0100	0100	0100	0010	2
0001	0001	0100	0010	2
0001	0001	0100	0100	2

Figura 4.18 Casos vecinos al nuevo caso.

Luego que nosotros encontramos, para el nuevo caso, los casos más cercanos o más parecido, la técnica requiere que se determine previamente la cantidad de casos vecinos con que se va a hacer la evaluación.

Es decir, si consideramos que el caso en mención se evalúe con respecto a su primer caso vecino, la salida al nuevo caso se le asocia la misma salida del este primer caso vecino. Por tanto para el nuevo caso la salida corresponde 0001.

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
0100	0001	0100	0100

Tabla 4.9 Salida del nuevo caso considerando primer caso vecino

Pero, de acuerdo a las transformaciones considerada 0001 corresponde a un rango de 80 a 90, lo cual esta es la salida o predicción final.

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
90	60	87	80-90
0100	0001	0100	0100

Tabla 4.10 Salida final.

Ahora consideremos, que para éste mismo caso, se evaluará su salida ahora considerando a sus primeros 4 casos vecinos.

De la misma forma que el anterior, su salida dependerá de estos casos. En la siguiente grafica podemos observar los 4 casos vecinos referentes al nuevo caso a predecir.

Casos Vecinos				
Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica	Atributos Iguales
0100	0001	0100	0001	3
0100	0001	1000	0100	2
0100	0001	0010	0010	2
0100	0001	0000	0010	2
0100	0100	0100	0010	2
0100	0100	0100	0010	2
0001	0001	0100	0010	2
0001	0001	0100	0100	2

Figura 4.19 Considerando 4 casos vecinos

Teniendo los casos vecinos del nuevo caso, observamos que existen valores que se repiten en la salida (0010). Por tanto la salida para los nuevos casos va a depender de las salidas más comunes existentes en nuestros casos vecinos a considerar, la salida para nuestro nuevo caso es considerando 4 casos vecinos se muestra en la siguiente tabla y la transformación correspondiente de la salida de la predicción final.

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
90	60	87	80-90
0100	0001	0100	0100

Tabla 4.11 Salida al nuevo caso vecino.

Por lo tanto, para el caso que hemos considerado la predicción es que el estudiante aprobara la clase en un rango de calificación entre 80-90 puntos.

Las interfaces graficas para la introducción de un nuevo caso, la cantidad de vecinos a considerar, son mostradas a continuación:

Algoritmo Automatizado: Vecino Mas Cercano

PROGRAMACION GRÁFICA

Analisis y Procesamiento Evaluación Distancia Manhattan

Nuevo Caso

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
▶				

Cantidad de Vecinos: 2

Su transformación Booleana

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
*				

Casos Vecinos

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
*				




Figura 4.20 Interfaz de evaluación para un nuevo caso

4.4.9 Vecino Cercano y Distancia Manhattan.

Ahora consideramos la técnica de vecino más cercano utilizando el método llamado **Distancia Manhattan**. De igual forma las entradas son: el nuevo caso (Nota de las Asignaturas) y cantidad de vecino. El método consiste en calcular las distancias entre cada uno de los atributos para cada uno de los casos de acuerdo al nuevo caso.

Por ejemplo, considerando el nuevo caso:

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
90	60	87	?

Tabla 4.12 Nuevo caso

Y uno de los caso de nuestro conjunto de entrenamiento

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
83	65	100	84

Tabla 4.13 Caso de nuestro conjunto de entrenamiento

El cálculo de las distancias entre los atributos del nuevo caso y uno de los casos tomados de nuestro conjunto de entrenamiento donde la salida es **84** para la asignatura en la cual estamos haciendo la asignación es:

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
10	5	13	84

Tabla 4.14 Distancia del nuevo caso y el caso de nuestro conjunto de entrenamiento.

La interfaz grafica que muestra el cálculo de las distancias de los **155** casos de nuestro conjunto de entrenamiento es la siguiente:

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
▶		67		91
				100
	71	44	74	90
	0		68	80
	40		100	80
	83	65	100	84
	60	80	100	84
	69	84	100	80
	61	50		76
	0		100	90
	79	65	77	0
	60	93	100	100
	77	60	77	75
	66	88	62	5
	0		94	60
	65	63	66	0

Figura 4.21 Conjunto de Entrenamiento.

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica	Total
▶	90	7	87	91	184
	90	60	87	100	237
	19	16	13	90	48
	90	60	19	80	169
	50	60	13	80	123
	7	5	13	84	25
	30	20	13	84	63
	21	24	13	80	58
	29	10	87	76	126
	90	60	13	90	163
	11	5	10	0	26
	30	33	13	100	76
	13	0	10	75	23
	24	28	25	5	77
	90	60	7	60	157

Figura 4.22 Cálculo de Distancias.

Luego cuando hemos realizado este cálculo procedemos a ordenar los casos que sumen menores distancias entre los atributos

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica	Total
	6	2	8	77	16
	9	5	7	69	21
	13	0	10	75	23
	8	10	6	82	24
	7	5	13	84	25
	11	5	10	0	26
	16	10	3	60	29
	2	15	13	86	30
	14	11	5	78	30
	24	5	3	82	32
	3	22	7	78	32
	5	29	0	76	34
	19	3	13	86	35
	16	19	1	76	36
	20	10	8	83	38

Figura 4.23 Orden de vecindad

De igual manera tenemos que considerar la cantidad de casos vecinos con los que se evaluarán los nuevos casos. Si consideramos que el nuevo caso se evaluara al primer caso vecino, que será el de **16** tomando en cuenta que es el que posee la menor distancia, la salida de nuestro nuevo caso corresponderá al de este caso vecino el cual es **77**.

Pero, si tomamos ahora **4** que corresponden al de menor distancia la salida corresponderá al promedio de la cantidad de vecinos a considerar.

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica	Total
	6	2	8	77	16
	9	5	7	69	21
	13	0	10	75	23
	8	10	6	82	24

Figura 4.24 los 4 vecinos más cercano

Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
90	60	87	76

Tabla 4.15 salida del nuevo caso.

Algoritmo Automatizado: Vecino Mas Cercano

PROGRAMACION GRÁFICA

Análisis y Procesamiento Evaluación Distancia Manhattan

Ordenar

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
▶				

Calcular Distancias Cantidad de Vecinos 2

Evaluar

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
*				

Figura 4.25 Interfaz de evaluación para nuevos casos

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

A continuación se muestran las interfaces gráficas en donde se introducen los nuevos casos, se calculan las distancias, se ordenan y se evalúan.

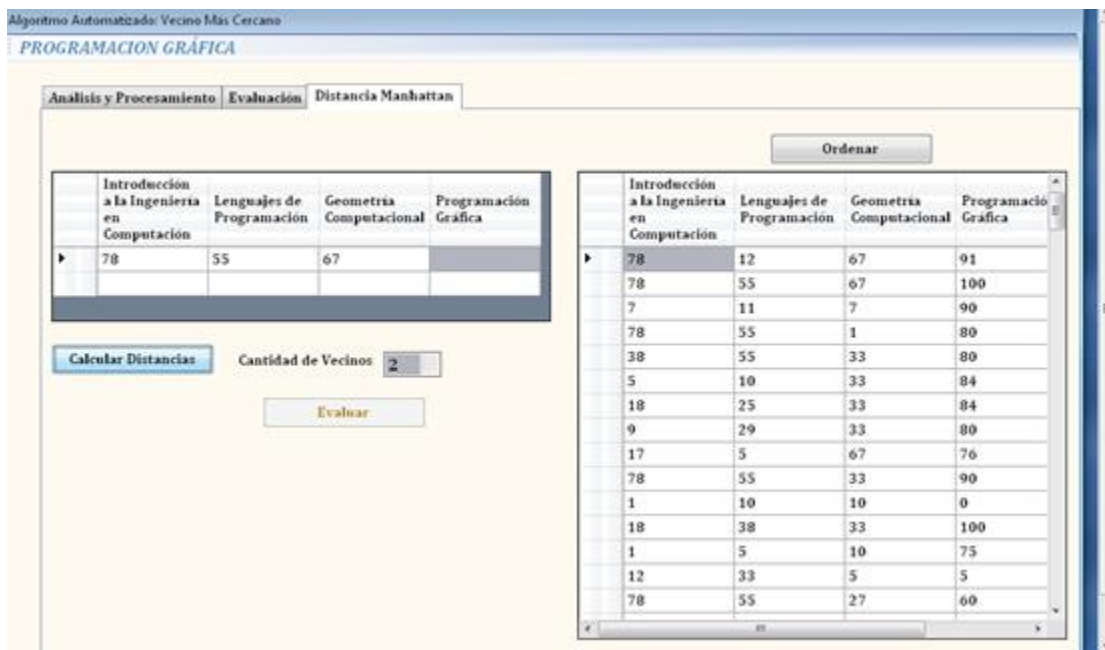


Figura 4.26 Encontrando vecinos más cercano.

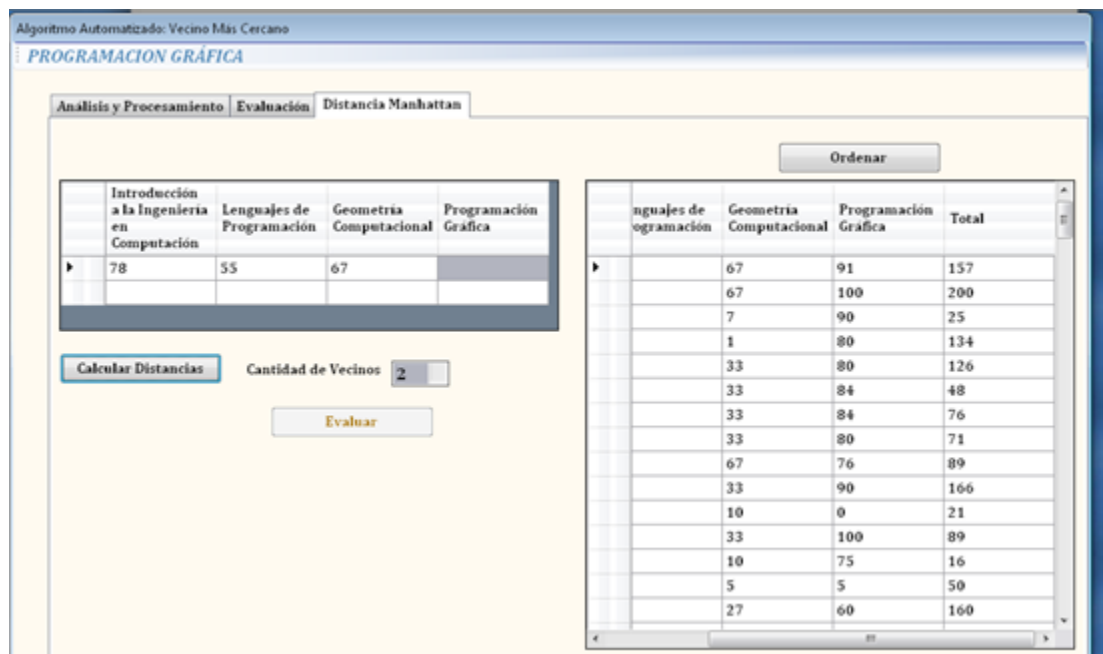


Figura 4.27 Indicando nuestra cantidad de vecinos

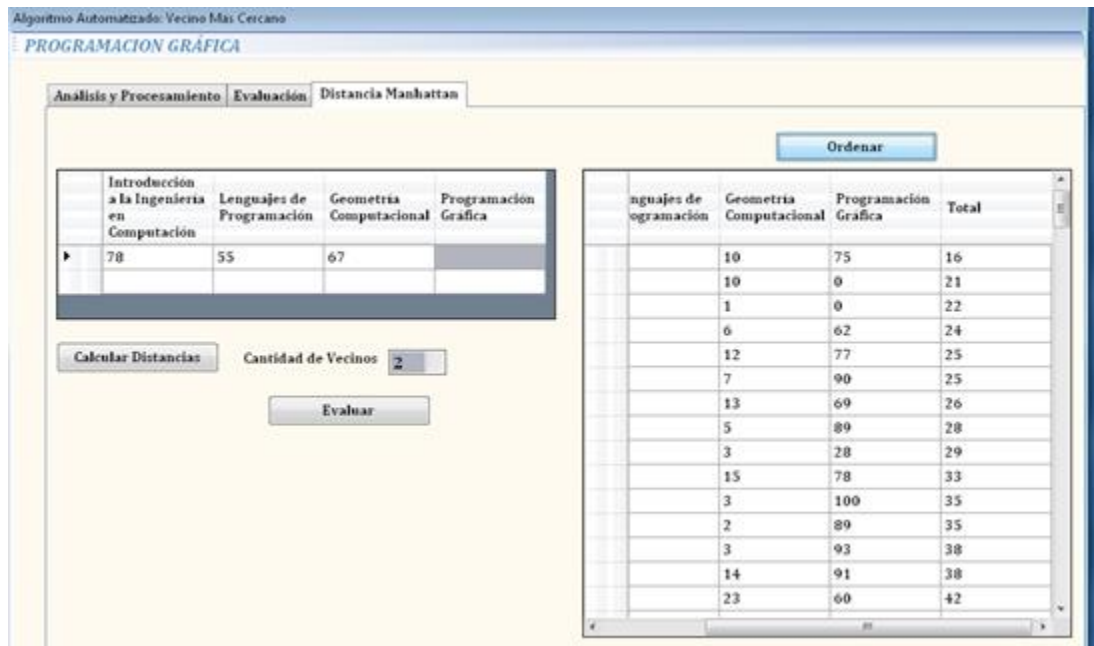


Figura 4.28 Ordenando nuestros casos más vecinos

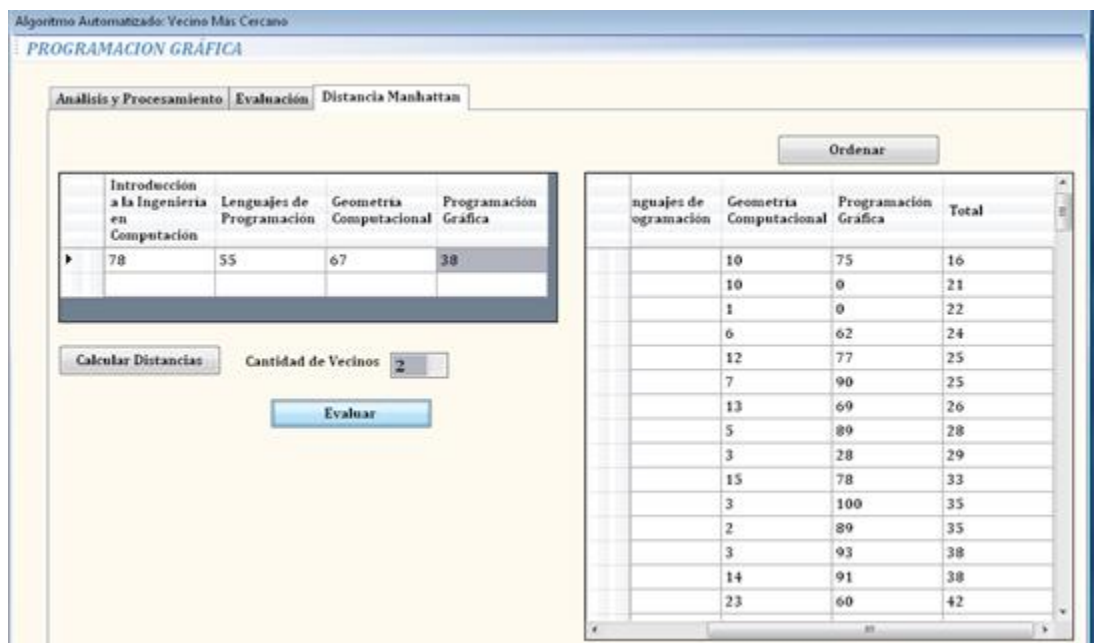


Figura 4.29 Evaluando nuestro nuevo caso

4.5 Validación Cruzada.

4.5.1 Propósito.

Utilizar validación cruzada para evaluar que tan bien trabaja un algoritmo de aprendizaje con un tipo de dato.

El objetivo principal de la técnica es saber que tan bien están trabajando los algoritmos de aprendizaje. En este caso hemos considerado evaluar que tan bien generan la salida las hipótesis que son generados por la técnica de Naive Bayes.

La técnica tiene el comportamiento que se muestra en la siguiente figura:

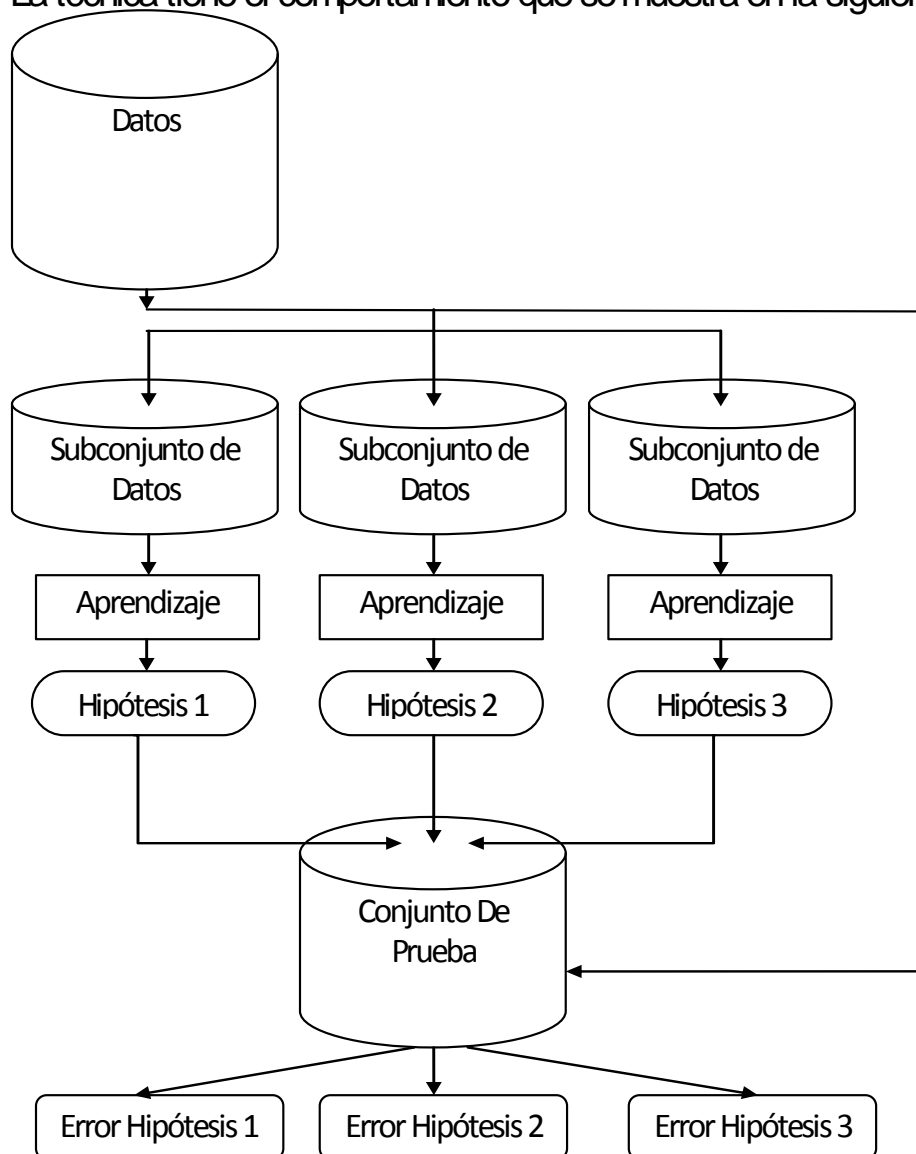


Figura 4.30 Representación de validación cruzada

Su trabajo básico es:

- ✓ Dado el conjunto de datos, separamos un subconjunto, el cual será nuestro conjunto de prueba.
- ✓ Los otros subconjuntos deberán ser sometidos a análisis con cualquiera de las técnicas de aprendizaje, seleccionando en este caso Naive Bayes
- ✓ De cada subconjunto se obtendrá una hipótesis la cual será enfrentada a nuestro conjunto de prueba.
- ✓ Como se tienen la salida de todos los casos del conjunto de prueba al evaluar con las diferentes hipótesis visualizaremos en cuantos casos falla.

4.5.3 Caso de Estudio.

- ✓ Se evaluará el algoritmo Naive Bayes utilizando validación cruzada, para la asignatura de Estadística.

Nos auxiliaremos del método de aprendizaje de Naive Bayes, considerando la asignatura de Estadística como nuestra salida y tomando como nuestros atributos las asignaturas de precedencia de la clase: las asignaturas son Matemáticas I, II, III.

De la misma manera que hemos procedido en los algoritmos anteriores, se cargan los datos, se realiza una transformación booleana. Tal y como se muestra en la siguiente figura:

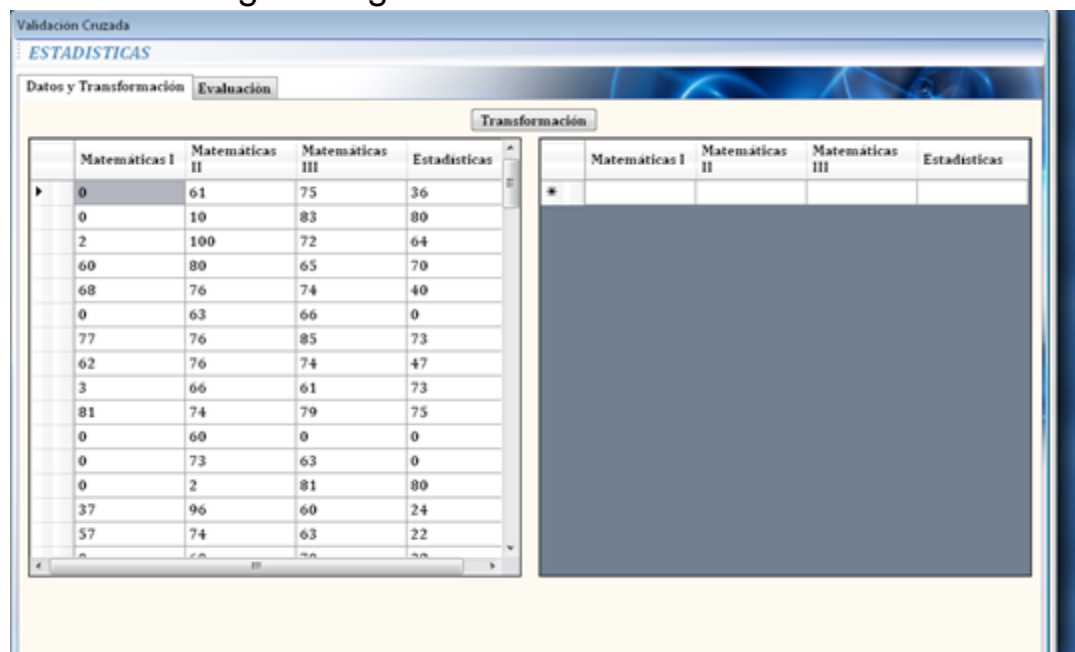


Figura 4.31 Selección de nuestro conjunto de entrenamiento

Procedemos a realizar la evaluación, generando 3 hipótesis en relación a 90 casos de 110, cada hipótesis es generada tomando en cuenta 30 casos, los 20 casos restantes de nuestro conjunto de entrenamiento será nuestro conjunto de pruebas en los que cada uno de los 20 casos se evaluara a cada una de las 3 hipótesis generadas, cuyo resultados se indican en la siguientes gráficas:

Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Transformación

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
0	61	75	36	
0	10	83	80	
2	100	72	64	
60	80	65	70	
68	76	74	40	
0	63	66	0	
77	76	85	73	
62	76	74	47	
3	66	61	73	
81	74	79	75	
0	60	0	0	
0	73	63	0	
0	2	81	80	
37	96	60	24	
57	74	63	22	

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
0	1	1	0	
0	0	1	1	
0	1	1	1	
1	1	1	1	
1	1	1	0	
0	1	1	0	
1	1	1	1	
1	1	1	0	
0	1	1	1	
0	1	0	0	
0	1	1	0	
0	0	1	1	
1	1	1	1	
0	1	0	0	
0	1	1	0	
0	1	1	1	
0	1	1	0	

Figura 4.32 transformación de nuestro conjunto de entrenamiento

Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Generar Hipótesis

	N*	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
*					

Conjunto de Prueba

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
*				

Evaluar Casos

	Matemáticas I	Matemáticas II	Matemáticas III	Estadística
*				

Figura 4.33 Interfaz de evaluación para nuevos casos

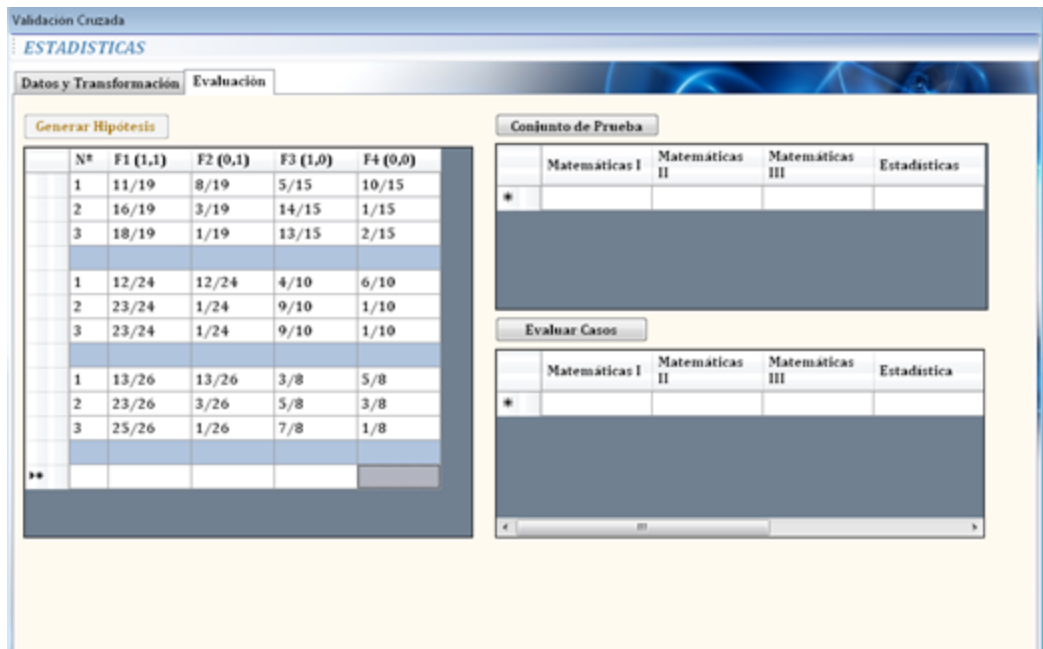


Figura 4.34 Encontrando nuestras hipótesis

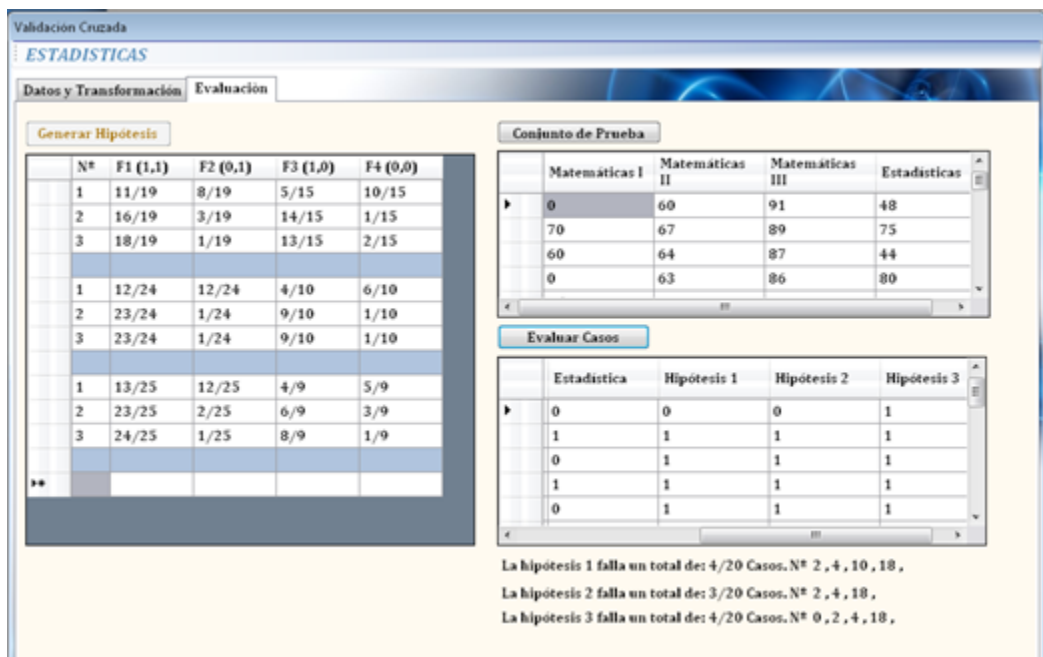


Figura 4.35 Selección de nuestro conjunto de prueba

En conclusión observamos que cada hipótesis que es evaluada a nuestro conjunto de prueba tiene un error promedio aproximadamente de 4 errores de los 20 casos totales, observamos además que cada hipótesis coincide al fallar en los casos 2, 4 y 18.

Representación Gráfica de Errores de la hipótesis con los 20 casos del conjunto de prueba.

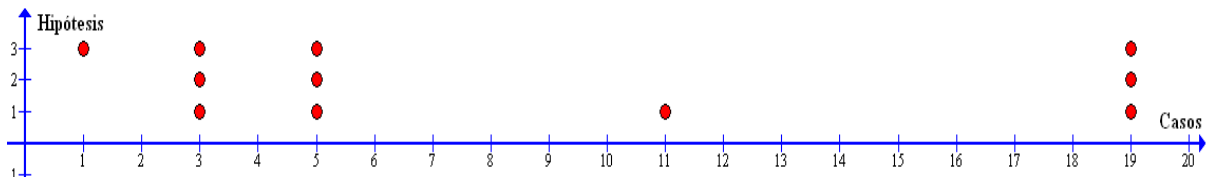


Figura 4.36 Representación gráfica de errores de nuestra hipótesis

Por tanto, la cantidad de errores por hipótesis es:

- La Hipótesis 1 tuvo un acierto de un 80% de los casos.
- La Hipótesis 2 tuvo un acierto de un 82% de los casos.
- La Hipótesis 3 tuvo un acierto de un 80% de los casos..

Conclusión.

Lo ideal es que se obtengan pocos errores al clasificar, en esta evaluación se tiene un error bajo. Permitiendo asegurar que la técnica de Naive Bayes se puede utilizar con seguridad para predecir quienes y cuantos estudiantes aprobaran la clase de Estadística.

CONCLUSIONES.

Para el desarrollo de este tema monográfico se investigó en sitios web y fuentes bibliográficas relacionada con el tema en cuestión, es importante mencionar que se establecieron y se desarrollaron distintos casos de estudios con el fin de implementar cada algoritmo y técnica, por tanto se hizo necesario analizar y estudiar la fuente de datos proporcionada por la División de informática y tecnología de la información (DITI-UNI).

Durante el desarrollo de la monografía se incorporaron conceptos sobre el campo de Aprendizaje de Máquina e inteligencia Artificial encontrados en OCW-MIT, con el propósito de cumplir con el objetivo de proporcionar una base de conocimientos que les permita a los interesados en el campo y como material de apoyo para la asignatura de Inteligencia Artificial.

Se ha logrado implementar y mostrar la aplicabilidad de los algoritmos de aprendizaje de máquina, si bien es cierto que se desarrollaron en una plataforma y gestor de base de datos genéricos se considera que la implementación de los algoritmos y técnicas de aprendizaje de máquina desarrollados en la monografía ayuda a la comprensión del mecanismos de trabajo de software especializado en esta área tales como PENTHO, RAPIDMINER pueden ser aplicados en otras herramientas software.

Con la implementación de las técnicas de aprendizaje de máquina sobre la base de datos de notas de los estudiantes de ingeniería en computación, se abre las puertas para realizar estudios de minería de datos con las notas de los estudiantes y que las autoridades puedan prever la cantidad de estudiantes que se tendrá en los cursos a iniciar.

Finalmente este programa promueve la investigación y utilización de técnicas y algoritmos de aprendizaje de máquina para analizar información en distintos áreas de estudio.

Se logró implementar 3 algoritmos y se aplicó la técnica de validación cruzada a uno de ellos. Estos hallazgos pueden servir para que un estudiante utilice el software IMEAN 1.0 para conocer un posible resultado a los casos de estudios considerados. Fomentar la necesidad de adquirir y conocer software que aborden el tema de minería de datos.

RECOMENDACIONES.

- ✓ Para implementar las distintas técnicas y algoritmos es importante documentarse sobre los casos de estudios con el fin de seleccionar la o las técnicas de aprendizaje de máquina y tener un conocimiento regular de programación lógica.
- ✓ Se recomienda promover en las asignaturas a fines, la utilidad de aprendizaje de máquina.
- ✓ Se recomienda a los representantes de Facultades promover trabajos interdisciplinarios que involucren a estudiantes de diferentes carreras de tal forma utilizar las técnicas de aprendizaje de máquina.
- ✓ Adquirir software con licencia para tener acceso a la documentación.
- ✓ Continuar explorando las técnicas con otros tipos de datos (sector empresarial, sector industrial, sector agrícola)
- ✓ Promover cursos de capacitación sobre el tema a nivel de postgrado.

BIBLIOGRAFÍA

Inteligencia Artificial un enfoque Moderno, 2da Edición

Russell Stuart, Norvig Peter

España 2004

Editorial Prentice Hall Hispanoamérica S.A.

Otras fuentes de información

- ✓ Artificial Intelligence- MIT

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2006/>

Readings

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2006/readings/>

Página de Descarga

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2006/download-course-materials/>

- ✓ Rapid Miner.

Felix Jungermann, Information Extraction with RapidMiner.

[http://www.google.com/url?sa=t&source=web&cd=9&ved=0CD8QFjAI&url=http%3A%2F%2Fwww-ai.informatik.uni-dortmund.de%2FDOKUMENTE%2Fjungermann_2009a.pdf&rct=j&q=rapid miner%20filetype%3Apdf&ei=bLyuTMWjNsWAIAeQtM3kDw&usg=AFQjCNErFIXN_wcjb2GIF6zAa-5iGt3Qyg&cad=rja](http://www.google.com/url?sa=t&source=web&cd=9&ved=0CD8QFjAI&url=http%3A%2F%2Fwww-ai.informatik.uni-dortmund.de%2FDOKUMENTE%2Fjungermann_2009a.pdf&rct=j&q=rapid+miner%20filetype%3Apdf&ei=bLyuTMWjNsWAIAeQtM3kDw&usg=AFQjCNErFIXN_wcjb2GIF6zAa-5iGt3Qyg&cad=rja)

Anexos

Cap. 6

ANEXOS

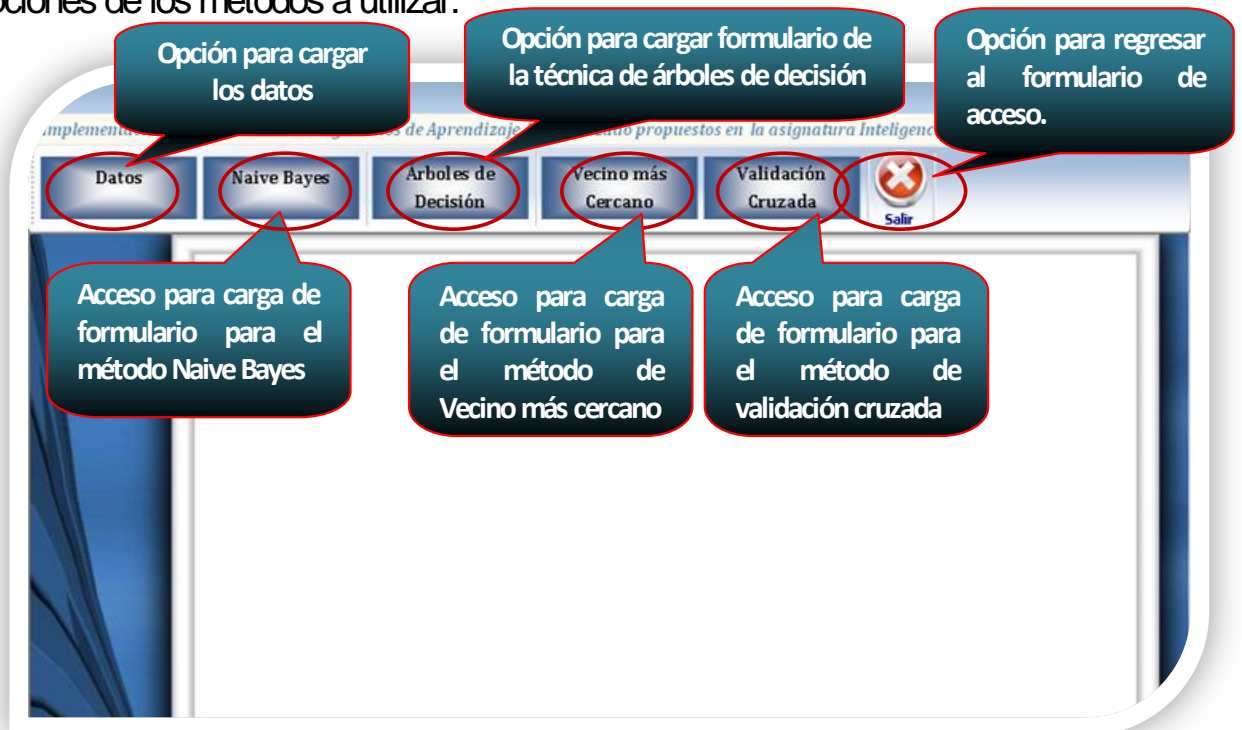
Anexo A: Manual de usuario e interfaces gráficas.

A continuación mostramos un manual de ayuda para el manejo de la aplicación y sus principales interfaces

1. Pantallas principales



Una vez que se acceda al programa la pantalla mostrar contiene una barra de opciones de los métodos a utilizar.



2. Carga y organización de Datos.

Cuando se selecciona la opción de “Datos”, el formulario carga un formulario correspondiente cuyas opciones básicas se especifican en la gráfica a continuación

Conjunto de Datos llamados desde SQL

Organización de datos para el tratamiento de cada método

Organización de datos

Carnet	Asignatura	Nota	Situación
2006-PPCPT	FISICA II	68	Regular
2006-PPCPT	PROGRAMACI...	91	Regular
2006-PPCPT	SISTEMAS DI...	65	Regular
2006-PPCPT	TEORIA DE L...	60	Regular
2006-PPCPT	ADMINISTRA...	90	Regular
2006-PPCPT	ARQUITECTU...	76	Regular
2006-PPCPT	ECONOMIA D...	72	Regular

	Matemáticas I	Conceptos	Lenguajes de Programación	Inglés II	Redacción Técnica	Matemáticas II	P00	Geometría	Sociología y
0		81	88	26	80	60	84		
0		0			0				
0		0			0				
0		6	34		0				0
0		1	0	40	0				
60		92	92	69		60	75	100	70
0		77			64	0	70		0
0		60	8	66		0	70		0

3. Interfaces para evaluación del método de aprendizaje Naive Bayes.

La interfaz de abajo aparece una vez seleccionada la opción “Naive Bayes”, sus opciones básicas se especifican en la gráfica

Pestaña para la generación de hipótesis

Pestaña que permite la transformación de datos

Pestaña que permite la evaluación de nuevos casos

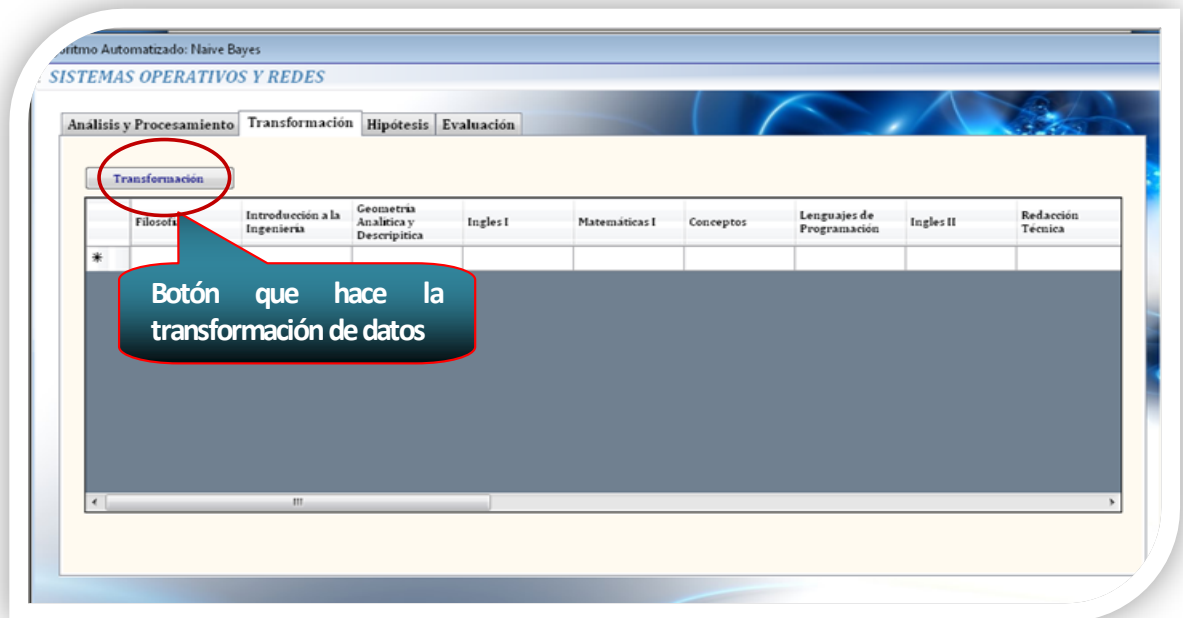
Total de casos considerados

Total 48/292

	Geometría	Inglés I	Inglés II	Redacción Técnica
65				
70				
61		60	64	84
60		77	86	93
62		65	60	65
81		78	71	71
0		74	79	77
65		80	81	60
67		71	70	97
65		88	68	83

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

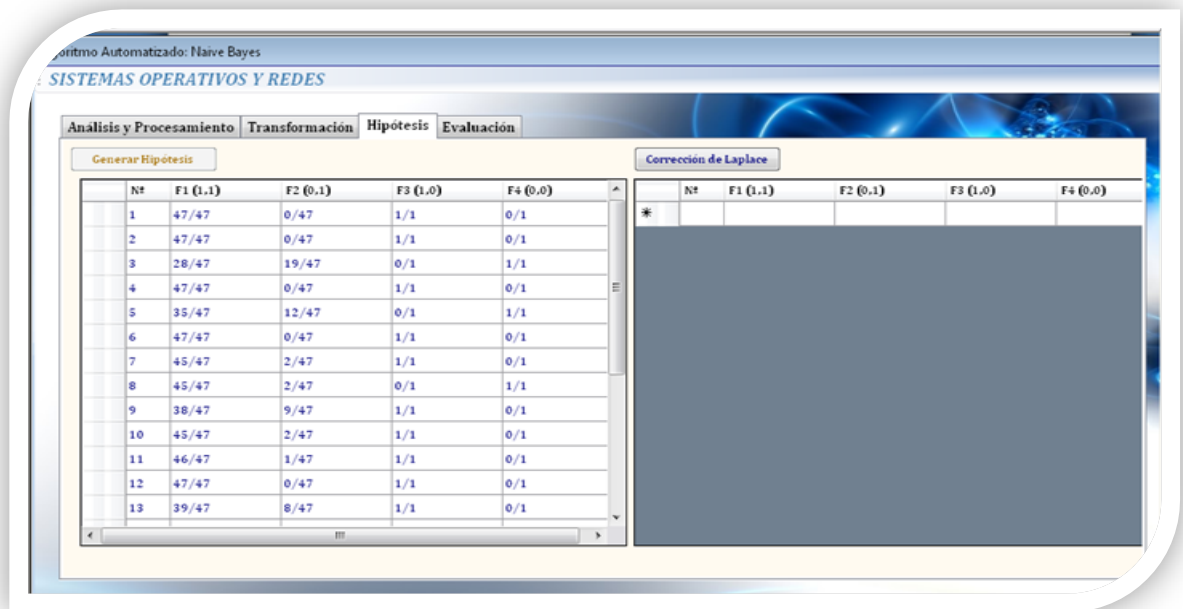
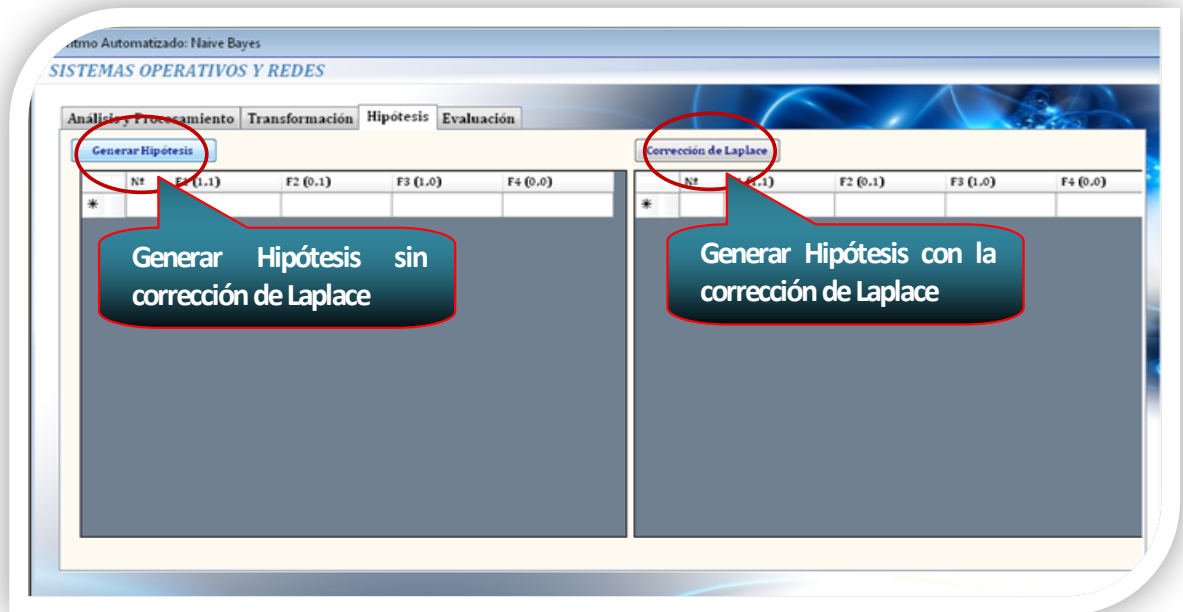
Cuando se selecciona la pestaña transformación, ésta nos permite realizar la transformación de datos. Ver figura de abajo



	Filosofía	Introducción a la Ingeniería	Geometría Analítica y Descriptiva	Inglés I	Matemáticas I	Conceptos	Lenguajes de Programación	Inglés II	Redacción Técnica
1	1	1	1	1	0	1	0	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1
1	1	1	0	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

Cuando se selecciona la pestaña de Hipótesis, ésta permite generar la hipótesis correspondiente al método de Naive Bayes



Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

Algoritmo Automatizado: Naive Bayes

SISTEMAS OPERATIVOS Y REDES

Analisis y Procesamiento Transformación Hipótesis Evaluación

Generar Hipótesis

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	47/47	0/47	1/1	0/1
2	47/47	0/47	1/1	0/1
3	28/47	19/47	0/1	1/1
4	47/47	0/47	1/1	0/1
5	35/47	12/47	0/1	1/1
6	47/47	0/47	1/1	0/1
7	45/47	2/47	1/1	0/1
8	45/47	2/47	0/1	1/1
9	38/47	9/47	1/1	0/1
10	45/47	2/47	1/1	0/1
11	46/47	1/47	1/1	0/1
12	47/47	0/47	1/1	0/1
13	39/47	8/47	1/1	0/1

Corrección de Laplace

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	48/49	1/49	2/3	1/3
2	48/49	1/49	2/3	1/3
3	29/49	20/49	1/3	2/3
4	48/49	1/49	2/3	1/3
5	36/49	13/49	1/3	2/3
6	48/49	1/49	2/3	1/3
7	46/49	3/49	2/3	1/3
8	46/49	3/49	1/3	2/3
9	39/49	10/49	2/3	1/3
10	46/49	3/49	2/3	1/3
11	47/49	2/49	2/3	1/3
12	48/49	1/49	2/3	1/3
13	40/49	9/49	2/3	1/3

Luego de la transformación y la generación de la hipótesis seleccionamos la pestaña evaluación cuya opciones se especifican en la siguiente gráfica

Algoritmo Automatizado: Naive Bayes

SISTEMAS OPERATIVOS Y REDES

Analisis y Procesamiento Transformación Evaluación

Introducción del nuevo caso

Evaluar un caso con la corrección de Laplace

Evaluar S/Corrección Laplace

Evaluar C/Corrección Laplace

S1

S2

Evaluar un caso sin la corrección de Laplace

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

Entorno Automatizado: Naive Bayes

SISTEMAS OPERATIVOS Y REDES

Analisis y Procesamiento | Transformación | Hipótesis | Evaluación

	Filosofía	Introducción a la Ingeniería	Geometría Analítica y Descriptiva	Inglés I	Matemáticas I	Conceptos	Lenguajes de Programación	Inglés II	Redacción Técnica
▶	65	88	100	75	100	0	74	48	49

◀ ▶

S1
S2

Entorno Automatizado: Naive Bayes

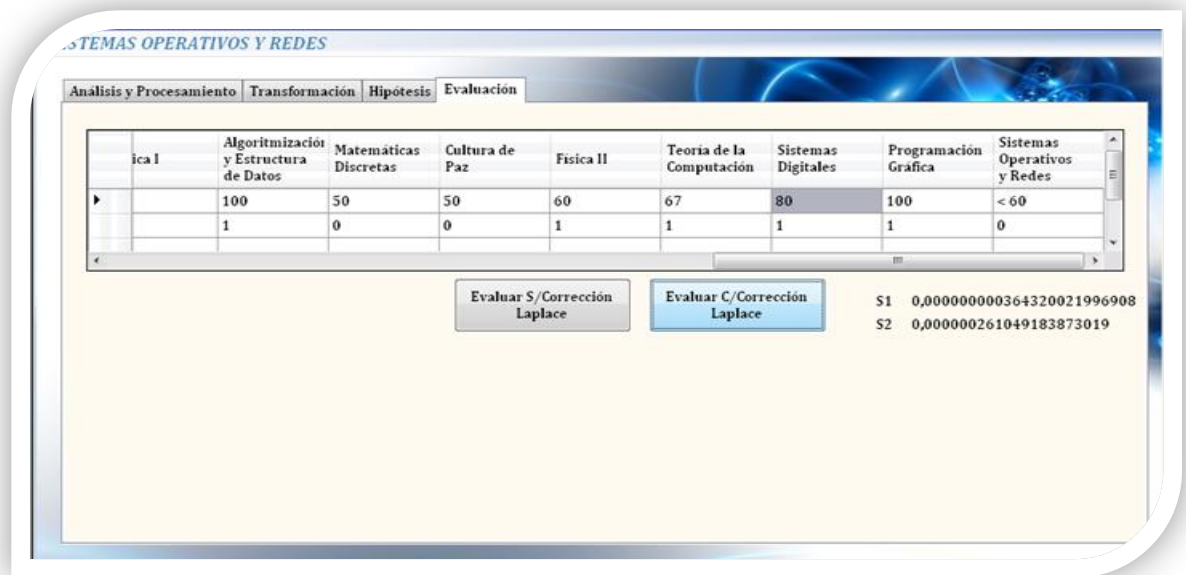
SISTEMAS OPERATIVOS Y REDES

Analisis y Procesamiento | Transformación | Hipótesis | Evaluación

	ica I	Algoritmización y Estructura de Datos	Matemáticas Discretas	Cultura de Paz	Física II	Teoría de la Computación	Sistemas Digitales	Programación Gráfica	Sistemas Operativos y Redes
▶		100	50	50	60	67	80	100	< 60
		1	0	0	1	1	1	1	0

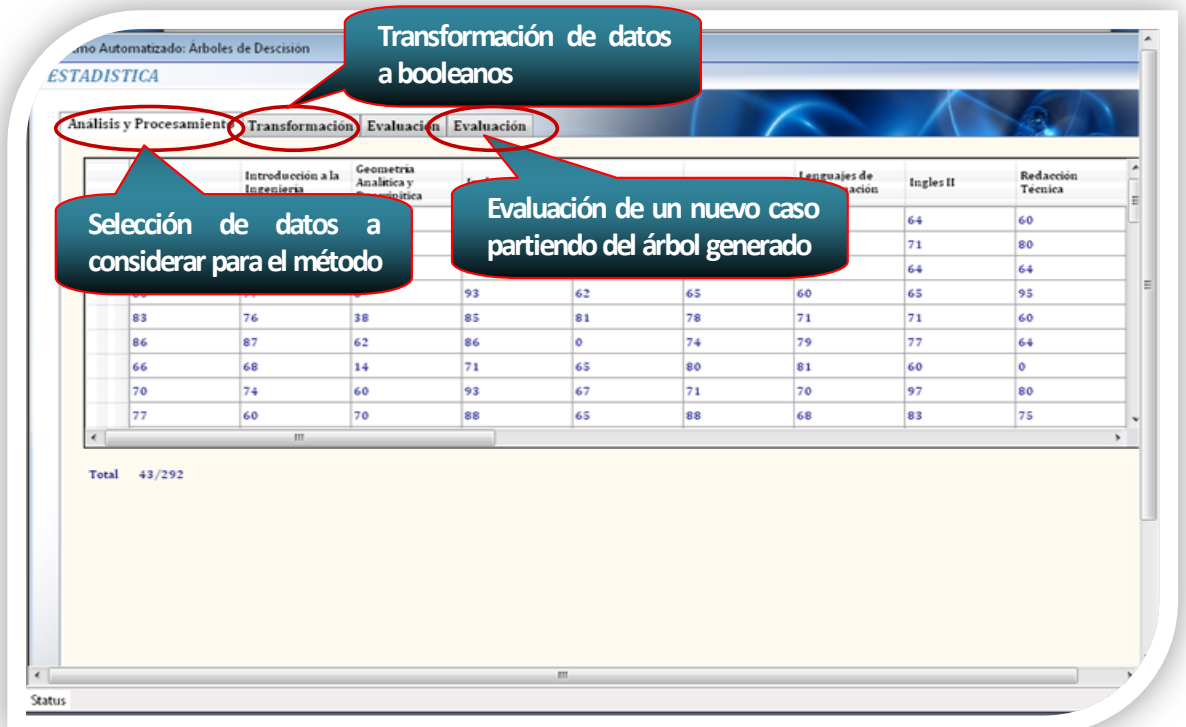
◀ ▶

S1 0
S2 0

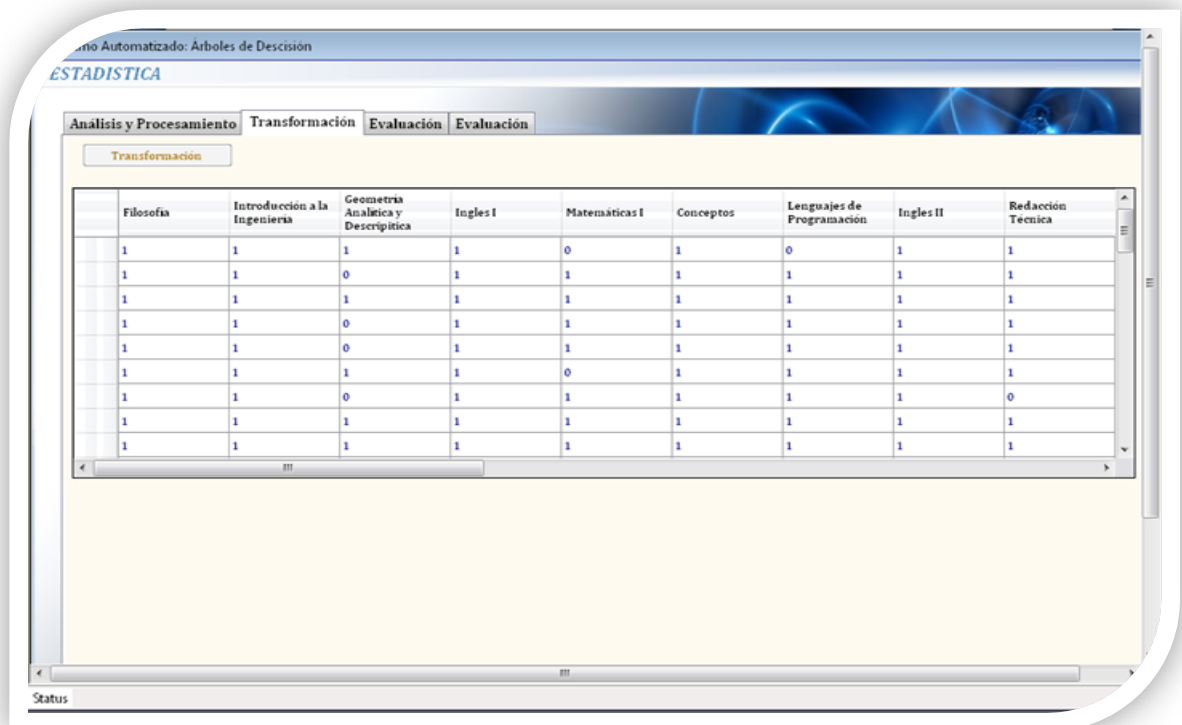
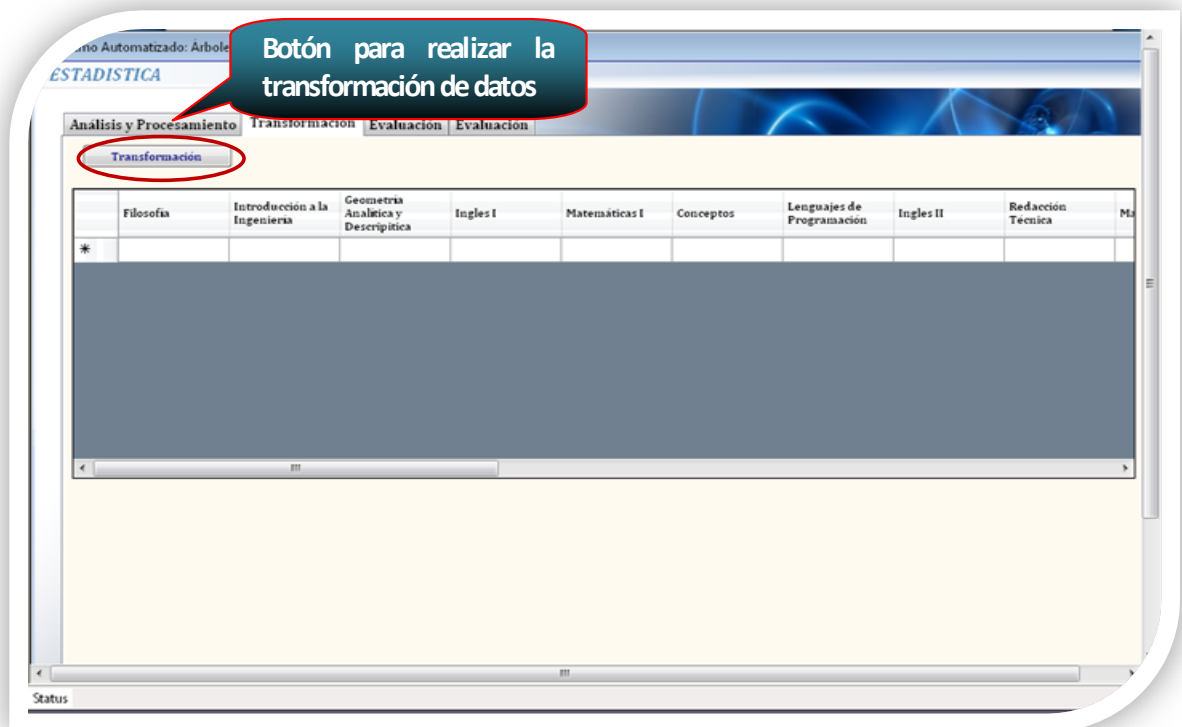


4. Interfaces para evaluación del método de aprendizaje Árboles de decisión

Cuando se selecciona la opción “Árboles de Decisión”, este es su formulario principal

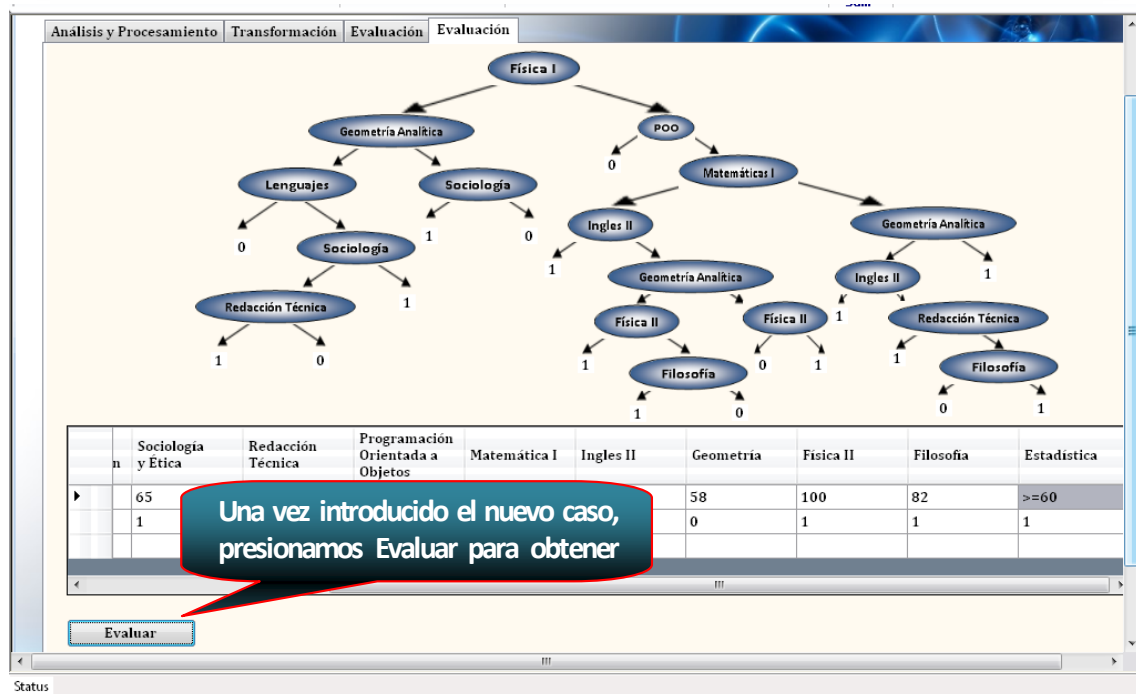


Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial



Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

Una vez cargado los datos, hecha la transformación y generado su hipótesis en forma de arbol evaluamos casos



5. Interfaces para evaluación del método de aprendizaje Naive Bayes

Cuando seleccionamos de la barra de opciones “Vecinos más cercano”, esta es la interfaz principal del método y sus diferentes opciones

The screenshot shows the main interface of the Naive Bayes evaluation tool. It features a menu bar with 'Análisis y Procesamiento', 'Evaluación', and 'Distancia Manhattan'. Below the menu is a 'Transformación' button. The main area contains two tables. The left table, titled 'Evaluación de un nuevo caso', displays numerical data for four categories: 'Introducción a la Ingeniería en Computación', 'Lenguajes de Programación', 'Geometría Computacional', and 'Programación Gráfica'. The right table, titled 'Transformación de datos', is currently empty. Callouts with red borders point to the 'Conjunto de datos' menu item, the 'Evaluación utilizando distancias Manhattan' menu item, the 'Evaluación' menu item, the 'Transformación' button, and the 'Evaluación de un nuevo caso' table.

Conjunto de datos

Evaluación utilizando distancias Manhattan

Análisis y Procesamiento Evaluación Distancia Manhattan

Transformación

Evaluación de un nuevo caso

Transformación de datos

This screenshot shows the same interface as the previous one, but with binary data (0s and 1s) in the tables. The left table, 'Evaluación de un nuevo caso', shows binary values for the same four categories. The right table, 'Transformación de datos', now displays binary data for the same categories. The 'Transformación' button is highlighted in yellow. The status bar at the bottom indicates 'Total 155'.

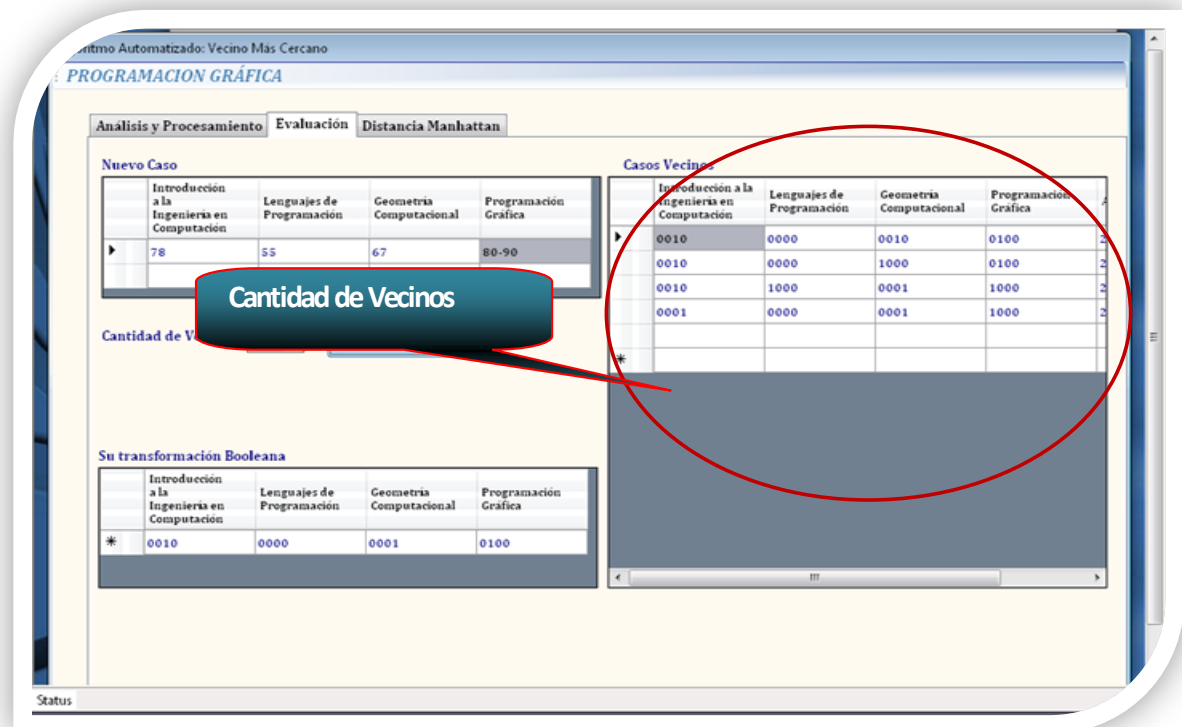
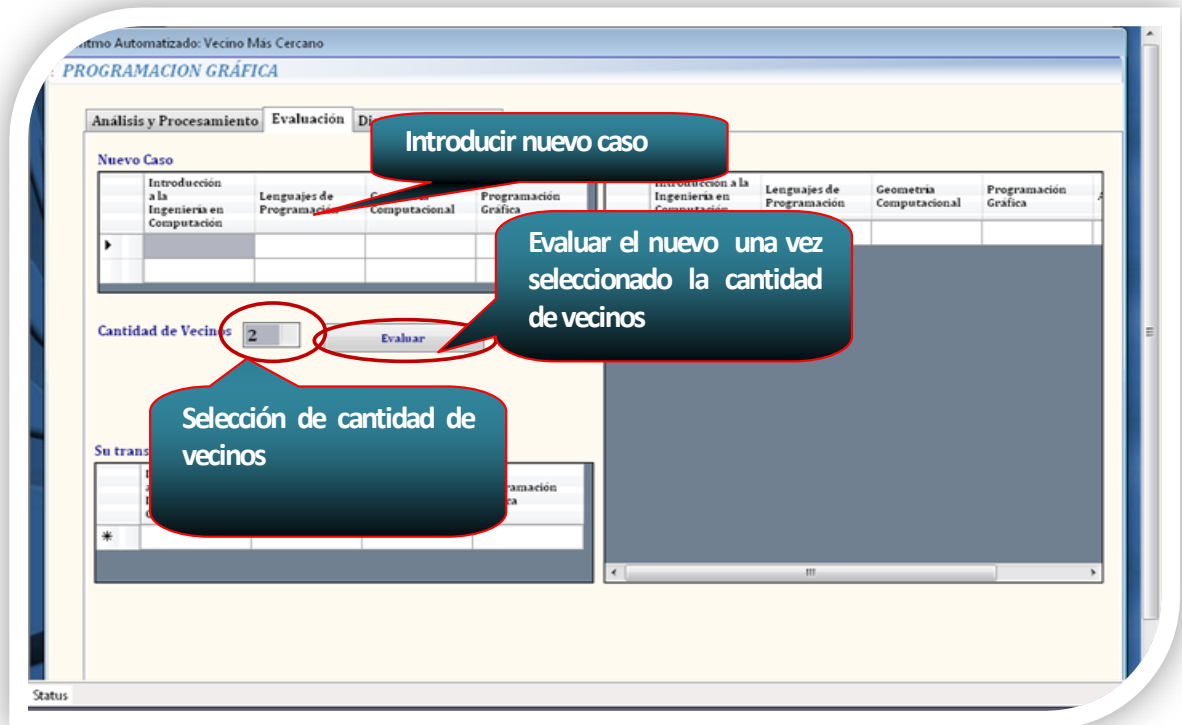
PROGRAMACION GRÁFICA

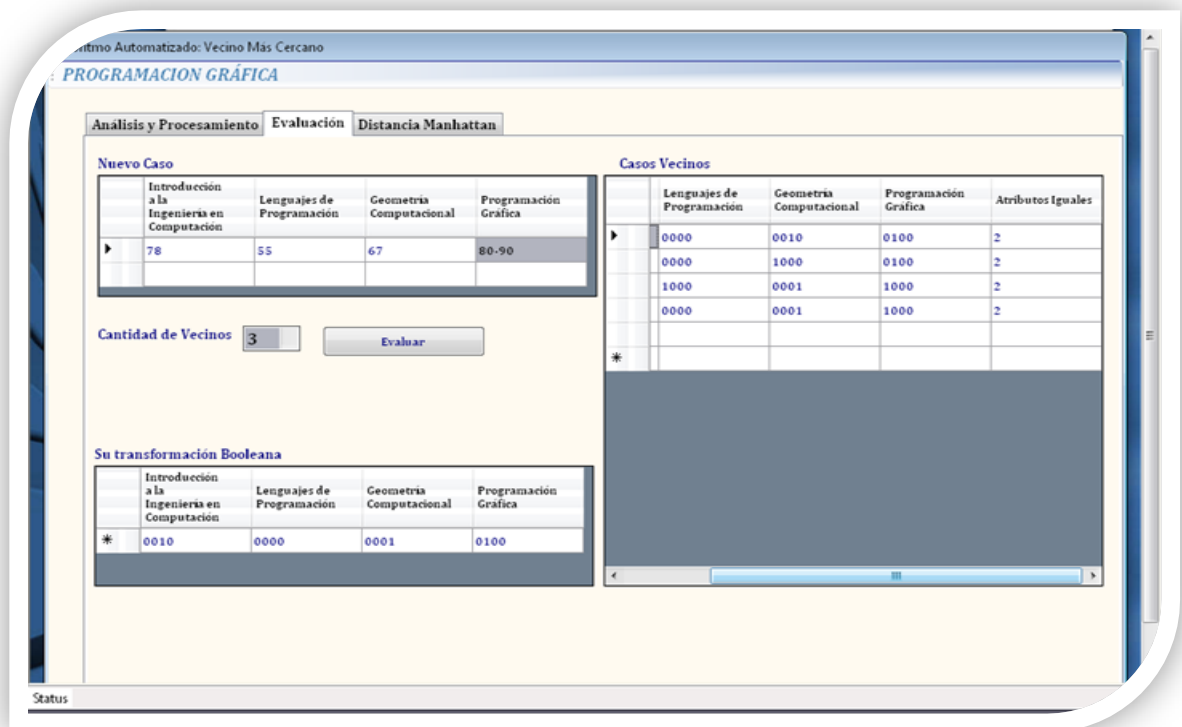
Análisis y Procesamiento Evaluación Distancia Manhattan

Transformación

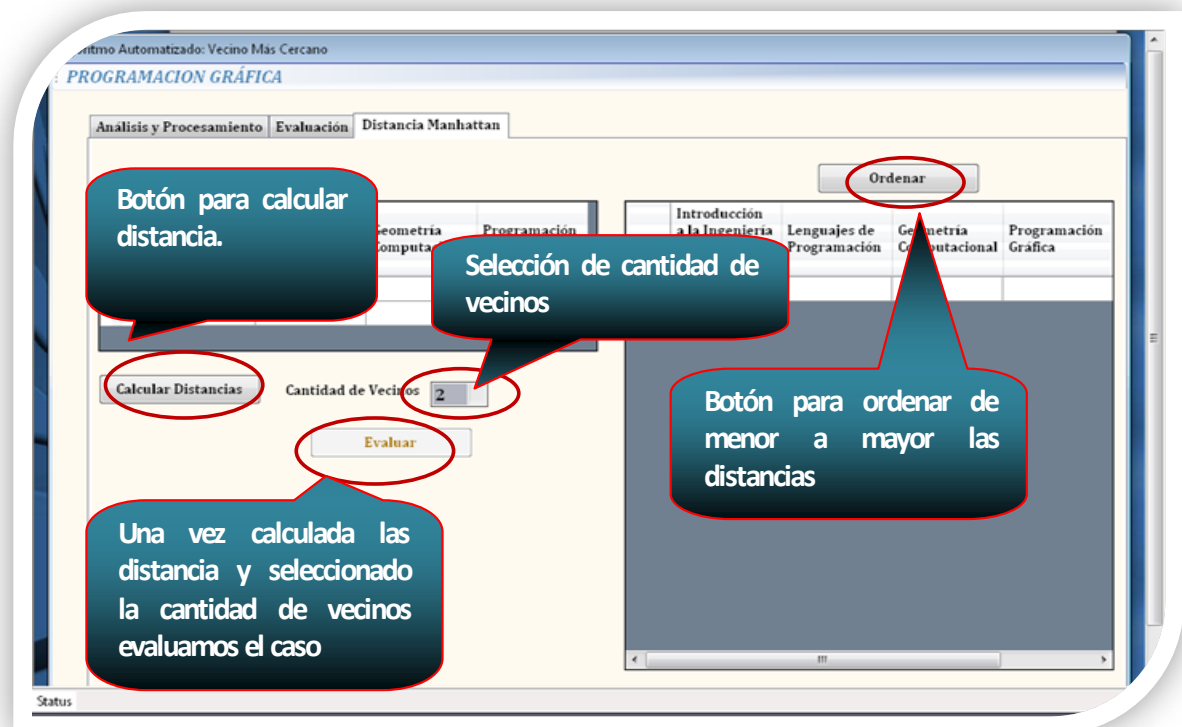
Evaluación de un nuevo caso

Transformación de datos





La siguiente interfaz muestra la técnica de distancia Manhattan para aplicar el método de vecino más cercano



Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

Algoritmo Automatizado: Vecino Más Cercano

PROGRAMACION GRÁFICA

Analisis y Procesamiento Evaluación Distancia Manhattan

Ordenar

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
▶	78	55	67	

Calcular Distancias Cantidad de Vecinos 2

Evaluar

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
*				

Status

Algoritmo Automatizado: Vecino Más Cercano

PROGRAMACION GRÁFICA

Analisis y Procesamiento Evaluación Distancia Manhattan

Ordenar

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
▶	78	55	67	

Calcular Distancias Cantidad de Vecinos 2

Evaluar

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
▶	78	12	67	91
	78	55	67	100
	7	11	7	90
	78	55	1	80
	38	55	33	80
	5	10	33	84
	18	25	33	84
	9	29	33	80
	17	5	67	76
	78	55	33	90
	1	10	10	0
	18	38	33	100
	1	5	10	75
	12	33	5	5
	78	55	27	60

Status

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

Entorno Automatizado: Vecino Mas Cercano

PROGRAMACION GRÁFICA

Analisis y Procesamiento | Evaluación | Distancia Manhattan

Ordenar

	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
▶	78	55	67	

Calcular Distancias Cantidad de Vecinos 2

Evaluar

	Lenguajes de Programación	Geometría Computacional	Programación Gráfica	Total
▶		67	91	157
		67	100	200
		7	90	25
		1	80	134
		33	80	126
		33	84	48
		33	84	76
		33	80	71
		67	76	89
		33	90	166
		10	0	21
		33	100	89
		10	75	16
		5	5	50
		27	60	160

Status

Entorno Automatizado: Vecino Mas Cercano

PROGRAMACION GRÁFICA

Analisis y Procesamiento | Evaluación | Distancia Manhattan

Ordenar

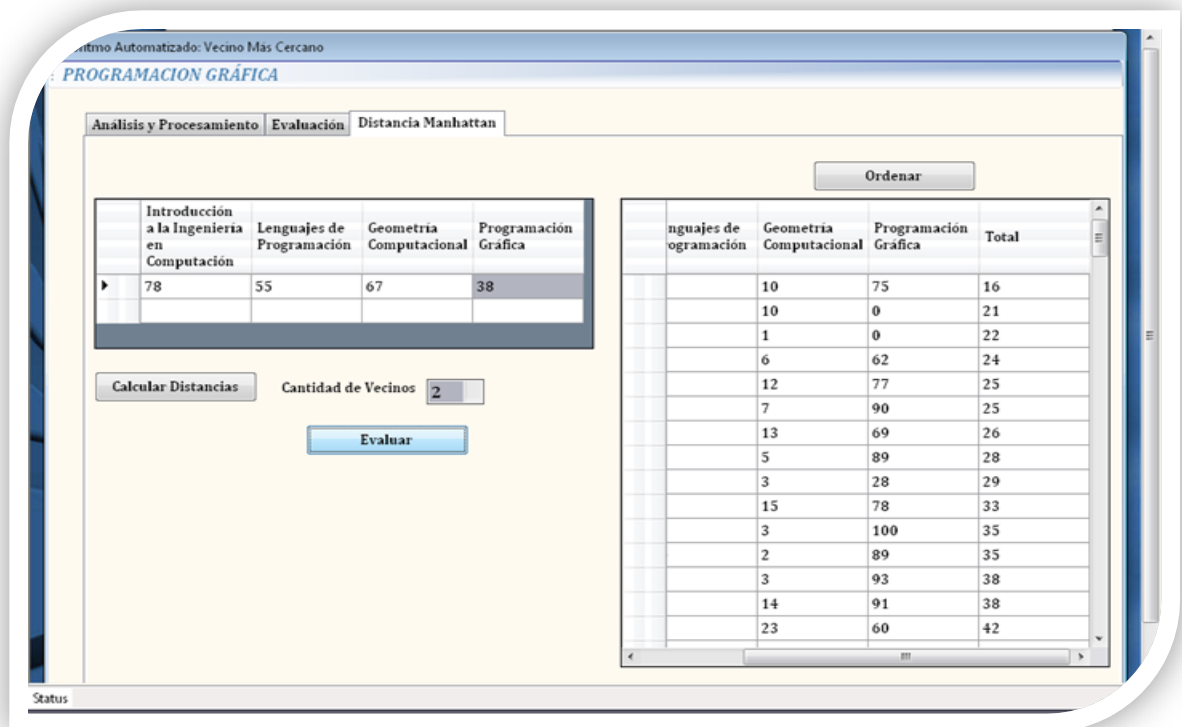
	Introducción a la Ingeniería en Computación	Lenguajes de Programación	Geometría Computacional	Programación Gráfica
▶	78	55	67	

Calcular Distancias Cantidad de Vecinos 2

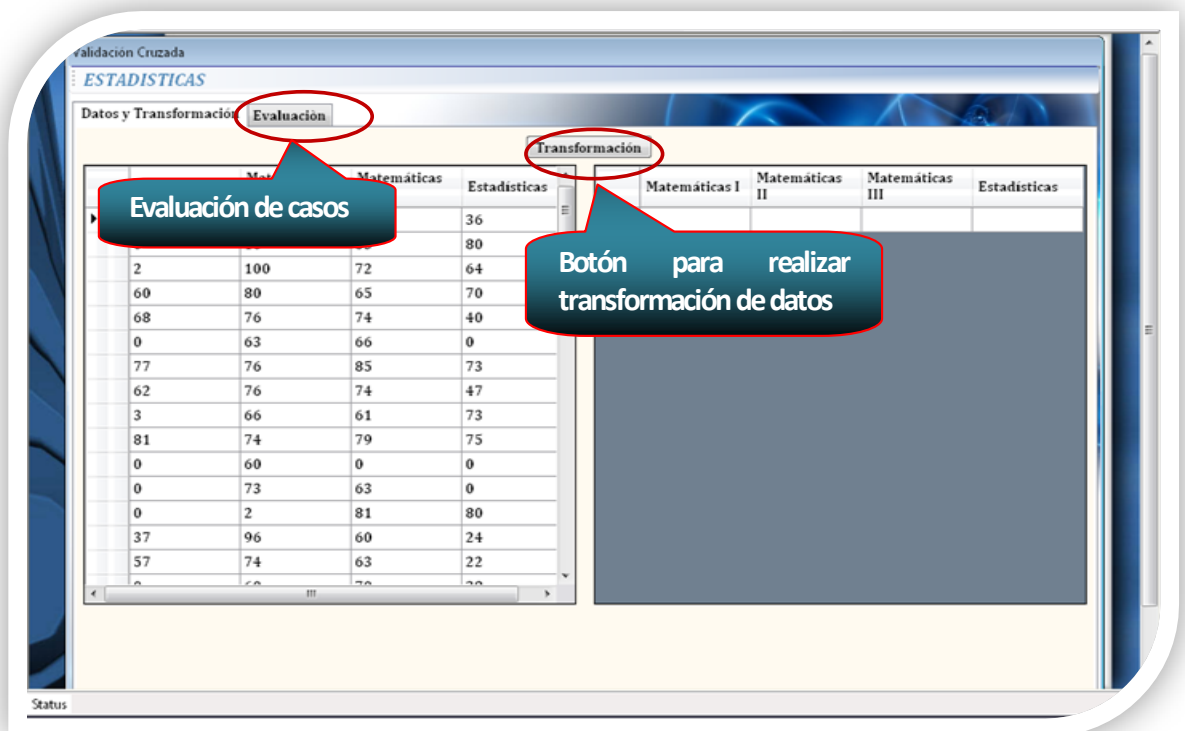
Evaluar

	Lenguajes de Programación	Geometría Computacional	Programación Gráfica	Total
		10	75	16
		10	0	21
		1	0	22
		6	62	24
		12	77	25
		7	90	25
		13	69	26
		5	89	28
		3	28	29
		15	78	33
		3	100	35
		2	89	35
		3	93	38
		14	91	38
		23	60	42

Status



6. **Validación Cruzada.** La grafica siguiente muestra la pantalla que muestra una vez que se haya dado de la barra de opciones Validación Cruzada.



Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Transformación

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
0	61	75	36	
0	10	83	80	
2	100	72	64	
60	80	65	70	
68	76	74	40	
0	63	66	0	
77	76	85	73	
62	76	74	47	
3	66	61	73	
81	74	79	75	
0	60	0	0	
0	73	63	0	
0	2	81	80	
37	96	60	24	
57	74	63	22	

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
0	1	1	0	
0	0	1	1	
0	1	1	1	
1	1	1	1	
1	1	1	1	
0	1	1	0	
1	1	1	1	
1	1	1	1	
0	1	1	1	
1	1	1	1	
0	1	0	0	
0	1	1	0	
0	0	1	1	
0	1	1	0	
0	1	1	0	

Status

Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Generar Hipótesis

Conjunto de Prueba

Evaluar Casos

Muestra el conjunto de prueba

Botón que genera la hipótesis

Botón para que genera la salida del conjunto de prueba

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
*				

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
*				

	Matemáticas I	Matemáticas II	Matemáticas III	Estadística
*				

Status

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Generar Hipótesis

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	11/19	8/19	5/15	10/15
2	16/19	3/19	14/15	1/15
3	18/19	1/19	13/15	2/15
1	12/24	12/24	4/10	6/10
2	23/24	1/24	9/10	1/10
3	23/24	1/24	9/10	1/10
1	13/26	13/26	3/8	5/8
2	23/26	3/26	5/8	3/8
3	25/26	1/26	7/8	1/8
»»				

Conjunto de Prueba

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
*				

Evaluar Casos

	Matemáticas I	Matemáticas II	Matemáticas III	Estadística
*				

Status

Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Generar Hipótesis

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	11/19	8/19	5/15	10/15
2	16/19	3/19	14/15	1/15
3	18/19	1/19	13/15	2/15
1	12/24	12/24	4/10	6/10
2	23/24	1/24	9/10	1/10
3	23/24	1/24	9/10	1/10
1	13/26	13/26	3/8	5/8
2	23/26	3/26	5/8	3/8
3	25/26	1/26	7/8	1/8
»»				

Conjunto de Prueba

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
72	75	86	80	
0	60	90	60	
40	82	89	60	
98	83	95	80	

Evaluar Casos

	Matemáticas I	Matemáticas II	Matemáticas III	Estadística
*				

Status

Implementación y evaluación de los algoritmos de Aprendizaje Automatizado propuestos en la asignatura Inteligencia Artificial

Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Generar Hipótesis

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	11/19	8/19	5/15	10/15
2	16/19	3/19	14/15	1/15
3	18/19	1/19	13/15	2/15
1	12/24	12/24	4/10	6/10
2	23/24	1/24	9/10	1/10
3	23/24	1/24	9/10	1/10
1	13/25	12/25	4/9	5/9
2	23/25	2/25	6/9	3/9
3	24/25	1/25	8/9	1/9

Conjunto de Prueba

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
0	60	91	48	
70	67	89	75	
60	64	87	44	
0	63	86	80	

Evaluar Casos

	Estadística	Hipótesis 1	Hipótesis 2	Hipótesis 3
0	0	0	0	1
1	1	1	1	1
0	1	1	1	1
1	1	1	1	1
0	1	1	1	1

La hipótesis 1 falla un total de: 4/20 Casos. Nº 2, 4, 10, 18,

La hipótesis 2 falla un total de: 3/20 Casos. Nº 2, 4, 18,

La hipótesis 3 falla un total de: 4/20 Casos. Nº 0, 2, 4, 18,

Status

Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Generar Hipótesis

Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	11/19	8/19	5/15	10/15
2	16/19	3/19	14/15	1/15
3	18/19	1/19	13/15	2/15
1	12/24	12/24	4/10	6/10
2	23/24	1/24	9/10	1/10
3	23/24	1/24	9/10	1/10
1	13/25	12/25	4/9	5/9
2	23/25	2/25	6/9	3/9
3	24/25	1/25	8/9	1/9

Conjunto de Prueba

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
0	60	91	48	
70	67	89	75	
60	64	87	44	
0	63	86	80	

Evaluar Casos

	Estadística	Hipótesis 1	Hipótesis 2	Hipótesis 3
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	1

La hipótesis 1 falla un total de: 4/20 Casos. Nº 2, 4, 10, 18,

La hipótesis 2 falla un total de: 3/20 Casos. Nº 2, 4, 18,

La hipótesis 3 falla un total de: 4/20 Casos. Nº 0, 2, 4, 18,

Status

Validación Cruzada

ESTADÍSTICAS

Datos y Transformación Evaluación

Generar Hipótesis

	Nº	F1 (1,1)	F2 (0,1)	F3 (1,0)	F4 (0,0)
1	11/19	8/19	5/15	10/15	
2	16/19	3/19	14/15	1/15	
3	18/19	1/19	13/15	2/15	
1	12/24	12/24	4/10	6/10	
2	23/24	1/24	9/10	1/10	
3	23/24	1/24	9/10	1/10	
1	13/25	12/25	4/9	5/9	
2	23/25	2/25	6/9	3/9	
3	24/25	1/25	8/9	1/9	
»»					

Conjunto de Prueba

	Matemáticas I	Matemáticas II	Matemáticas III	Estadísticas
0	60	91	48	
70	67	89	75	
60	64	87	44	
0	63	86	80	

Evaluar Casos

	Estadística	Hipótesis 1	Hipótesis 2	Hipótesis 3
1	0	1	1	
1	1	1	1	
1	1	1	1	
1	1	1	1	
1	1	1	1	

La hipótesis 1 falla un total de: 4/20 Casos. Nº 2, 4, 10, 18,

La hipótesis 2 falla un total de: 3/20 Casos. Nº 2, 4, 18,

La hipótesis 3 falla un total de: 4/20 Casos. Nº 0, 2, 4, 18,

Status

Anexo B: Código fuente de implementación

b.1 Datos

```
Imports System.Data.SqlClient

Public Class Datos
    Dim da As SqlDataAdapter
    Private ds As DataSet

    Private Sub Datos_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim c As New SqlConnection
        c = New SqlConnection
        c.ConnectionString = Settings1.Default.Cadena2
        da = New SqlDataAdapter("SELECT Carnet, Asignatura, Nota,
Modalidad FROM dbo.COMPUTACION$ WHERE Modalidad = 'Regular'", c)
        Dim oCommBuild As SqlCommandBuilder = New SqlCommandBuilder(da)
        c.Open()
        ds = New DataSet
        da.Fill(ds, "COMPUTACION$")
        c.Close()
        dgrDatos.DataSource() = ds
        dgrDatos.DataMember() = "COMPUTACION$"
        dgrDatos.Width = 520
        dgrDatos.Height = 176
    End Sub
```

Organización de Datos

```
Private Sub cmdOrganizacion_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdOrganizacion.Click
    Dim n, i, a As Integer
    Dim Carnet1, Carnet2, Carnet() As String
    n = dgrDatos.Rows.Count - 1
    Carnet2 = dgrDatos.Rows(i).Cells(0).Value
    ReDim Carnet(588)
    Carnet(a) = Carnet2
    dgrOrganizacion.Rows.Add()
    dgrOrganizacion.Rows(a).Cells(0).Value = Carnet(a)
    dgrOrganizacion.Rows.Add()
    For i = 0 To n - 1
        Carnet1 = dgrDatos.Rows(i).Cells(0).Value

        If (Carnet1 = Carnet2) Then
            'MsgBox(Trim(dgr.Rows(i).Cells(1)..Value))

            If Trim(dgrDatos.Rows(i).Cells(1).Value) = "FILOSOFIA"
Then
                dgrOrganizacion.Rows(a).Cells(1).Value =
dgrDatos.Rows(i).Cells(2).Value
            End If

            If Trim(dgrDatos.Rows(i).Cells(1).Value) =
"INTRODUCCION A INGENIERIA EN COMPUTACION" Then
                dgrOrganizacion.Rows(a).Cells(2).Value =
dgrDatos.Rows(i).Cells(2).Value
```

```
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) = "GEOMETRIA
ANALITICA Y DESCRIPTIVA" Then
    dgrOrganizacion.Rows(a).Cells(3).Value =
dgrDatos.Rows(i).Cells(2).Value
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) = "INGLES I"
Then
    dgrOrganizacion.Rows(a).Cells(4).Value =
dgrDatos.Rows(i).Cells(2).Value
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) = "MATEMATICA
I" Then
    dgrOrganizacion.Rows(a).Cells(5).Value =
dgrDatos.Rows(i).Cells(2).Value
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) = "CONCEPTOS
DE LENGUAJES" Then
    dgrOrganizacion.Rows(a).Cells(6).Value =
dgrDatos.Rows(i).Cells(2).Value
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) = "LENGUAJE DE
PROGRAMACION" Then
    dgrOrganizacion.Rows(a).Cells(7).Value =
dgrDatos.Rows(i).Cells(2).Value
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) = "INGLES II"
Then
    dgrOrganizacion.Rows(a).Cells(8).Value =
dgrDatos.Rows(i).Cells(2).Value
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) = "REDACCION
TECNICA" Then
    dgrOrganizacion.Rows(a).Cells(9).Value =
dgrDatos.Rows(i).Cells(2).Value
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) = "MATEMATICA
II" Then
    dgrOrganizacion.Rows(a).Cells(10).Value =
dgrDatos.Rows(i).Cells(2).Value
End If

If Trim(dgrDatos.Rows(i).Cells(1).Value) =
"PROGRAMACION ORIENTADA A OBJETO" Then
    dgrOrganizacion.Rows(a).Cells(11).Value =
dgrDatos.Rows(i).Cells(2).Value
End If
```

```

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "GEOMETRIA
COMPUTACIONAL" Then
            dgrOrganizacion.Rows (a) .Cells (12) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "SOCIOLOGIA
Y ETICA" Then
            dgrOrganizacion.Rows (a) .Cells (13) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "MATEMATICA
III" Then
            dgrOrganizacion.Rows (a) .Cells (14) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "FISICA I"
Then
            dgrOrganizacion.Rows (a) .Cells (15) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) =
"ALGORITMIZACION Y ESTRUCTURAS DE DATOS" Then
            dgrOrganizacion.Rows (a) .Cells (16) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "MATEMATICAS
DISCRETAS" Then
            dgrOrganizacion.Rows (a) .Cells (17) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "CULTURA DE
PAZ Y DERECHOS HUMANOS" Then
            dgrOrganizacion.Rows (a) .Cells (18) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "FISICA II"
Then
            dgrOrganizacion.Rows (a) .Cells (19) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "TEORIA DE
LA COMPUTACION" Then
            dgrOrganizacion.Rows (a) .Cells (20) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "SISTEMAS
DIGITALES" Then
```

```
dgrOrganizacion.Rows(a).Cells(21).Value =  
dgrDatos.Rows(i).Cells(2).Value  
End If  
  
If Trim(dgrDatos.Rows(i).Cells(1).Value) =  
"PROGRAMACION GRAFICA" Then  
dgrOrganizacion.Rows(a).Cells(22).Value =  
dgrDatos.Rows(i).Cells(2).Value  
End If  
  
If Trim(dgrDatos.Rows(i).Cells(1).Value) =  
"ESTADISTICA" Then  
dgrOrganizacion.Rows(a).Cells(23).Value =  
dgrDatos.Rows(i).Cells(2).Value  
End If  
  
If Trim(dgrDatos.Rows(i).Cells(1).Value) = "SISTEMAS  
OPERATIVOS Y REDES" Then  
dgrOrganizacion.Rows(a).Cells(24).Value =  
dgrDatos.Rows(i).Cells(2).Value  
End If  
  
If Trim(dgrDatos.Rows(i).Cells(1).Value) = "BASES DE  
DATOS" Then  
dgrOrganizacion.Rows(a).Cells(25).Value =  
dgrDatos.Rows(i).Cells(2).Value  
End If  
  
If Trim(dgrDatos.Rows(i).Cells(1).Value) = "ECONOMIA"  
Then  
dgrOrganizacion.Rows(a).Cells(26).Value =  
dgrDatos.Rows(i).Cells(2).Value  
End If  
  
If Trim(dgrDatos.Rows(i).Cells(1).Value) = "HARDWARE  
REPARACION Y MANTENIMIENTO" Then  
dgrOrganizacion.Rows(a).Cells(27).Value =  
dgrDatos.Rows(i).Cells(2).Value  
End If  
  
Else  
Carnet2 = Carnet1  
a = a + 1  
Carnet(a) = Carnet2  
'MsgBox(Carnet(a))  
dgrOrganizacion.Rows.Add()  
dgrOrganizacion.Rows(a).Cells(0).Value = Carnet(a)  
If Trim(dgrDatos.Rows(i).Cells(1).Value) = "FILOSOFIA"  
Then  
dgrOrganizacion.Rows(a).Cells(1).Value =  
dgrDatos.Rows(i).Cells(2).Value  
End If  
  
If Trim(dgrDatos.Rows(i).Cells(1).Value) =  
"INTRODUCCION A INGENIERIA EN COMPUTACION" Then
```

```
                dgrOrganizacion.Rows(a).Cells(2).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "GEOMETRIA  
ANALITICA Y DESCRIPTIVA" Then  
                    dgrOrganizacion.Rows(a).Cells(3).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "INGLES I"  
Then  
                    dgrOrganizacion.Rows(a).Cells(4).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "MATEMATICA  
I" Then  
                    dgrOrganizacion.Rows(a).Cells(5).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "CONCEPTOS  
DE LENGUAJES" Then  
                    dgrOrganizacion.Rows(a).Cells(6).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "LENGUAJE DE  
PROGRAMACION" Then  
                    dgrOrganizacion.Rows(a).Cells(7).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "INGLES II"  
Then  
                    dgrOrganizacion.Rows(a).Cells(8).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "REDACCION  
TECNICA" Then  
                    dgrOrganizacion.Rows(a).Cells(9).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "MATEMATICA  
II" Then  
                    dgrOrganizacion.Rows(a).Cells(10).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) =  
"PROGRAMACION ORIENTADA A OBJETO" Then  
                    dgrOrganizacion.Rows(a).Cells(11).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                    End If
```

```

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "GEOMETRIA
COMPUTACIONAL" Then
            dgrOrganizacion.Rows (a) .Cells (12) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "SOCIOLOGIA
Y ETICA" Then
            dgrOrganizacion.Rows (a) .Cells (13) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "MATEMATICA
III" Then
            dgrOrganizacion.Rows (a) .Cells (14) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "FISICA I"
Then
            dgrOrganizacion.Rows (a) .Cells (15) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) =
"ALGORITMIZACION Y ESTRUCTURAS DE DATOS" Then
            dgrOrganizacion.Rows (a) .Cells (16) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "MATEMATICAS
DISCRETAS" Then
            dgrOrganizacion.Rows (a) .Cells (17) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "CULTURA DE
PAZ Y DERECHOS HUMANOS" Then
            dgrOrganizacion.Rows (a) .Cells (18) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "FISICA II"
Then
            dgrOrganizacion.Rows (a) .Cells (19) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "TEORIA DE
LA COMPUTACION" Then
            dgrOrganizacion.Rows (a) .Cells (20) .Value =
dgrDatos.Rows (i) .Cells (2) .Value
        End If

        If Trim(dgrDatos.Rows (i) .Cells (1) .Value) = "SISTEMAS
DIGITALES" Then
```



```
                dgrOrganizacion.Rows(a).Cells(21).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) =  
"PROGRAMACION GRAFICA" Then  
                    dgrOrganizacion.Rows(a).Cells(22).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) =  
"ESTADISTICA" Then  
                    dgrOrganizacion.Rows(a).Cells(23).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "SISTEMAS  
OPERATIVOS Y REDES" Then  
                    dgrOrganizacion.Rows(a).Cells(24).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "BASES DE  
DATOS" Then  
                    dgrOrganizacion.Rows(a).Cells(25).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "ECONOMIA"  
Then  
                    dgrOrganizacion.Rows(a).Cells(26).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                End If  
  
                If Trim(dgrDatos.Rows(i).Cells(1).Value) = "HARDWARE  
REPARACION Y MANTENIMIENTO" Then  
                    dgrOrganizacion.Rows(a).Cells(27).Value =  
dgrDatos.Rows(i).Cells(2).Value  
                End If  
            End If  
        Next  
    End Sub  
End Class
```

b.2 Naive Bayes

b.2.1 Selección De Datos

```
Private Sub TSNB Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TSNB.Click
    ChildForm1.Show()
    ChildForm2.Hide()
    ChildForm3.Hide()
    ChildForm4.Hide()
    ChildForm5.Hide()

    Dim i, j, total, cont, a As Integer
    Dim var As String
    total = Me.ChildForm4.dgrOrganizacion.Rows.Count - 1
    a = 0
    Me.ChildForm1.dgrDatosNB.Rows.Add()
    For i = 0 To total - 2
        For j = 1 To 22
            If
IsDBNull(Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(j).Value) Then
                cont = cont + 1
            Else
                var =
Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(j).Value
                If var = "" Then
                    cont = cont + 1
                End If
            End If
        Next j
        If (cont = 0) Then
            For j = 0 To 21
                Me.ChildForm1.dgrDatosNB.Rows(a).Cells(j).Value =
Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(j + 1).Value
            Next j
            Me.ChildForm1.dgrDatosNB.Rows(a).Cells(j).Value =
Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(24).Value
            a = a + 1
            Me.ChildForm1.dgrDatosNB.Rows.Add()
        End If
        cont = 0
    Next i
    Me.ChildForm1.TotalCasos11.Text = a &
Me.ChildForm1.TotalCasos11.Text

End Sub
```

b.2.2 Transformación de Datos

```
Public Class NB
    Dim cantidad(,) As Integer
    Dim positivos, negativos As Integer
    Private Sub Transformacion Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Transformacion.Click
        Dim i, j, TCasos As Integer
```

```
Transformacion.Enabled = False
TCasos = dgrDatosNB.Rows.Count - 3
dgSistemasO.Rows.Add()
For i = 0 To TCasos
    For j = 0 To 22
        If dgrDatosNB.Rows(i).Cells(j).Value >= 60 Then
            dgSistemasO.Rows(i).Cells(j).Value = 1
        Else
            dgSistemasO.Rows(i).Cells(j).Value = 0
        End If
    Next j
    dgSistemasO.Rows.Add()
Next i
End Sub
```

b.2.3 Generacion de Hipótesis sin la corrección de Laplace

```
Private Sub cmdGenerar_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdGenerar.Click
    Dim i, j, TCasos, cont1, cont2, cont3, cont4 As Integer
    cmdGenerar.Enabled = False
    TCasos = dgSistemasO.Rows.Count - 3
    ReDim cantidad(22, 4)
    For i = 0 To TCasos
        If dgSistemasO.Rows(i).Cells(22).Value = 1 Then
            positivos = positivos + 1
        Else
            negativos = negativos + 1
        End If
    Next
    For j = 0 To 21
        For i = 0 To TCasos
            If dgSistemasO.Rows(i).Cells(j).Value = 1 And
dgSistemasO.Rows(i).Cells(22).Value = 1 Then
                cont1 = cont1 + 1
            ElseIf dgSistemasO.Rows(i).Cells(j).Value = 0 And
dgSistemasO.Rows(i).Cells(22).Value = 1 Then
                cont2 = cont2 + 1
            ElseIf dgSistemasO.Rows(i).Cells(j).Value = 1 And
dgSistemasO.Rows(i).Cells(22).Value = 0 Then
                cont3 = cont3 + 1
            ElseIf dgSistemasO.Rows(i).Cells(j).Value = 0 And
dgSistemasO.Rows(i).Cells(22).Value = 0 Then
                cont4 = cont4 + 1
            End If
        Next i
        cantidad(j, 0) = cont1
        cantidad(j, 1) = cont2
        cantidad(j, 2) = cont3
        cantidad(j, 3) = cont4
        cont1 = 0
        cont2 = 0
        cont3 = 0
        cont4 = 0
    Next j
    For i = 0 To 21
```

```
Hipotesis.Rows.Add()  
Next  
  
For i = 0 To 21  
    Hipotesis.Rows(i).Cells(0).Value = i + 1  
    Hipotesis.Rows(i).Cells(1).Value = cantidad(i, 0) & "/" &  
positivos  
    Hipotesis.Rows(i).Cells(2).Value = cantidad(i, 1) & "/" &  
positivos  
    Hipotesis.Rows(i).Cells(3).Value = cantidad(i, 2) & "/" &  
negativos  
    Hipotesis.Rows(i).Cells(4).Value = cantidad(i, 3) & "/" &  
negativos  
Next  
End Sub
```

b.2.4 Generación de Hipótesis con la corrección de Laplace

```
Private Sub cmdCLaplace Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles cmdCLaplace.Click  
    Dim i As Integer  
    cmdCLaplace.Enabled = False  
  
    For i = 0 To 21  
        HipotesisL.Rows.Add()  
    Next  
    For i = 0 To 21  
        HipotesisL.Rows(i).Cells(0).Value = i + 1  
        HipotesisL.Rows(i).Cells(1).Value = cantidad(i, 0) + 1 &  
"/" & positivos + 2  
        HipotesisL.Rows(i).Cells(2).Value = cantidad(i, 1) + 1 &  
"/" & positivos + 2  
        HipotesisL.Rows(i).Cells(3).Value = cantidad(i, 2) + 1 &  
"/" & negativos + 2  
        HipotesisL.Rows(i).Cells(4).Value = cantidad(i, 3) + 1 &  
"/" & negativos + 2  
    Next  
End Sub
```

b.2.5 Evaluación de un nuevo caso sin la corrección de Laplace

```
Private Sub cmdSCorreccion Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles cmdSCorreccion.Click  
    Dim i As Integer  
    Dim s1, s2 As Decimal  
    s1 = 1 : s2 = 1  
    dgCaso.Rows.Add()  
    For i = 0 To 21  
        If dgCaso.Rows(0).Cells(i).Value >= 60 Then  
            dgCaso.Rows(1).Cells(i).Value = 1  
        Else  
            dgCaso.Rows(1).Cells(i).Value = 0  
        End If  
    Next  
  
    For i = 0 To 21  
        If dgCaso.Rows(1).Cells(i).Value = 1 Then  
            s1 = s1 * (cantidad(i, 0) / positivos)
```

```
Else
    s1 = s1 * (cantidad(i, 1) / positivos)
End If
Next i

For i = 0 To 21
    If dgCaso.Rows(1).Cells(i).Value = 1 Then
        s2 = s2 * (cantidad(i, 2) / negativos)
    Else
        s2 = s2 * (cantidad(i, 3) / negativos)
    End If
Next i
If s1 > s2 Then
    dgCaso.Rows(1).Cells(22).Value = 1
    dgCaso.Rows(0).Cells(22).Value = ">= 60"
Else
    dgCaso.Rows(1).Cells(22).Value = 0
    dgCaso.Rows(0).Cells(22).Value = "< 60"
End If
V1.Text = s1
V2.Text = s2
End Sub
```

b.2.6 Evaluación de un nuevo caso con la corrección de Laplace

```
Private Sub cmdCCorreccion_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCCorreccion.Click
```

```
    Dim i As Integer
    Dim s1, s2 As Decimal
    s1 = 1 : s2 = 1
    dgCaso.Rows.Add()
    For i = 0 To 21
        If dgCaso.Rows(0).Cells(i).Value >= 60 Then
            dgCaso.Rows(1).Cells(i).Value = 1
        Else
            dgCaso.Rows(1).Cells(i).Value = 0
        End If
    Next

    For i = 0 To 21
        If dgCaso.Rows(1).Cells(i).Value = 1 Then
            s1 = s1 * ((cantidad(i, 0) + 1) / (positivos + 2))
        Else
            s1 = s1 * ((cantidad(i, 1) + 1) / (positivos + 2))
        End If
    Next i

    For i = 0 To 21
        If dgCaso.Rows(1).Cells(i).Value = 1 Then
            s2 = s2 * ((cantidad(i, 2) + 1) / (negativos + 2))
        Else
            s2 = s2 * ((cantidad(i, 3) + 1) / (negativos + 2))
        End If
    Next i
    If s1 > s2 Then
        dgCaso.Rows(1).Cells(22).Value = 1
        dgCaso.Rows(0).Cells(22).Value = ">= 60"
```

```
Else  
    dgCaso.Rows (1) .Cells (22) .Value = 0  
    dgCaso.Rows (0) .Cells (22) .Value = "< 60"  
End If  
V1.Text = s1  
V2.Text = s2  
End Sub
```

b.3 Arbol de Decisión

b.3.1 Selección de Datos

```
Private Sub TSADD_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TSADD.Click
    ChildForm2.Show()
    ChildForm1.Hide()
    ChildForm3.Hide()
    ChildForm4.Hide()
    ChildForm5.Hide()
    Dim i, j, total, cont, a As Integer
    Dim var As String
    total = ChildForm4.dgrOrganizacion.Rows.Count - 1
    a = 0
    ChildForm2.dgrDatosAD.Rows.Add()
    For i = 0 To total - 2
        For j = 1 To 23
            If
IsDBNull (ChildForm4.dgrOrganizacion.Rows(i).Cells(j).Value) Then
                'cont = cont + 1
                ChildForm4.dgrOrganizacion.Rows(i).Cells(j).Value =
0
            Else
                var =
ChildForm4.dgrOrganizacion.Rows(i).Cells(j).Value
                If var = "" Then
                    cont = cont + 1
                End If
            End If
        Next j
        If (cont = 0) Then
            For j = 0 To 22
                ChildForm2.dgrDatosAD.Rows(a).Cells(j).Value =
ChildForm4.dgrOrganizacion.Rows(i).Cells(j + 1).Value
            Next j
            a = a + 1
            ChildForm2.dgrDatosAD.Rows.Add()
        End If
        cont = 0
    Next i
    ChildForm2.TotalCasos10.Text = a & ChildForm2.TotalCasos10.Text

End Sub
```

b.3.2 Transformación de Datos

```
Private Sub Transformacion_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Transformacion.Click
    Dim i, j, TCasos As Integer
    Transformacion.Enabled = False
    TCasos = dgrDatosAD.Rows.Count - 3
    dgBasesDatos.Rows.Add()
    For i = 0 To TCasos
        For j = 0 To 22
            If dgrDatosAD.Rows(i).Cells(j).Value >= 60 Then
                dgBasesDatos.Rows(i).Cells(j).Value = 1
            Else
```



```
dgBasesDatos.Rows(i).Cells(j).Value = 0
End If
Next j
dgBasesDatos.Rows.Add()
Next i
End Sub
```

b.3.3 Cálculo de Entropías

```
Private Sub cmdEntropia_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdEntropia.Click
```

```
Dim i, j, TCasos, cont1, cont2, cont3, cont4, Positivos, Negativos, Posicion, a, b As Integer
```

```
Dim H1, H2, P1, P2, AE(21), max As Decimal
```

```
TextBox1.Text = ""
```

```
TCasos = dgBasesDatos.Rows.Count - 3
```

```
For i = 0 To TCasos
```

```
    If dgBasesDatos.Rows(i).Cells(22).Value = 1 Then
```

```
        Positivos = Positivos + 1
```

```
    Else
```

```
        Negativos = Negativos + 1
```

```
    End If
```

```
Next
```

```
max = 10
```

```
For j = 0 To 21
```

```
    For i = 0 To TCasos
```

```
        If dgBasesDatos.Rows(i).Cells(j).Value = 0 Then
```

```
            If dgBasesDatos.Rows(i).Cells(22).Value = 1 Then
```

```
                cont1 = cont1 + 1
```

```
            Else
```

```
                cont2 = cont2 + 1
```

```
            End If
```

```
        Else
```

```
            If dgBasesDatos.Rows(i).Cells(22).Value = 1 Then
```

```
                cont3 = cont3 + 1
```

```
            Else
```

```
                cont4 = cont4 + 1
```

```
            End If
```

```
        End If
```

```
    Next i
```

```
    If (cont1 + cont2) > 0 Then
```

```
        P1 = cont1 / (cont1 + cont2)
```

```
    End If
```

```
    If (cont3 + cont4) > 0 Then
```

```
        P2 = cont3 / (cont3 + cont4)
```

```
    End If
```

```
    MsgBox(P1)
```

```
    MsgBox(P2)
```

```
    If (P1 <> 0) And (P1 <> 1) Then
```

```
        H1 = (-P1 * ((Math.Log10(P1)) / (Math.Log10(2)))) -
```

```
((1 - P1) * ((Math.Log10(1 - P1)) / (Math.Log10(2))))
```

```
    End If
```

```
    If (P2 <> 0) And (P2 <> 1) Then
```

```
        H2 = (-P2 * ((Math.Log10(P2)) / (Math.Log10(2)))) -
```

```
((1 - P2) * ((Math.Log10(1 - P2)) / (Math.Log10(2))))
```

```
    End If
```

```
AE(j) = (((cont1 + cont2) / TCasos) * H1) + (((cont3 +
cont4) / TCasos) * H2)

'MsgBox (H1)
'MsgBox (H2)
cont1 = 0
cont2 = 0
cont3 = 0
cont4 = 0
P1 = 0
P2 = 0
H1 = 0
H2 = 0

TextBox1.Text = TextBox1.Text & AE(j) & " - " & "(" & j + 1
& ")"

If AE(j) < max Then
    max = AE(j)
    Posicion = j
End If
Next j
MsgBox (max)
MsgBox (Posicion)
dgrP.Rows.Add()
dgrN.Rows.Add()
For i = 0 To TCasos
    If dgBasesDatos.Rows(i).Cells(14).Value = 0 Then
        For j = 0 To 22
            dgrP.Rows(a).Cells(j).Value =
dgBasesDatos.Rows(i).Cells(j).Value
        Next j
        dgrP.Rows.Add()
        a = a + 1
    Else
        For j = 0 To 22
            dgrN.Rows(b).Cells(j).Value =
dgBasesDatos.Rows(i).Cells(j).Value
        Next j
        dgrN.Rows.Add()
        b = b + 1
    End If
Next i

End Sub

Private Sub cmdEPromP Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdEPromP.Click
    Dim i, j, TCasos, cont1, cont2, cont3, cont4, Positivos,
Negativos, Posicion As Integer
    Dim H1, H2, P1, P2, AE(21), max As Decimal
    TextBox1.Text = ""
    TCasos = dgrP.Rows.Count - 3
    For i = 0 To TCasos
        If dgrP.Rows(i).Cells(22).Value = 1 Then
            Positivos = Positivos + 1
```

```
Else
    Negativos = Negativos + 1
End If
Next
max = 10
For j = 0 To 21
    For i = 0 To TCasos
        If dgrP.Rows(i).Cells(j).Value = 0 Then
            If dgrP.Rows(i).Cells(22).Value = 1 Then
                cont1 = cont1 + 1
            Else
                cont2 = cont2 + 1
            End If
        Else
            If dgrP.Rows(i).Cells(22).Value = 1 Then
                cont3 = cont3 + 1
            Else
                cont4 = cont4 + 1
            End If
        End If
    Next i
    If (cont1 + cont2) > 0 Then
        P1 = cont1 / (cont1 + cont2)
    End If
    If (cont3 + cont4) > 0 Then
        P2 = cont3 / (cont3 + cont4)
    End If

    'MsgBox (P1)
    'MsgBox (P2)
    If (P1 <> 0) And (P1 <> 1) Then
        H1 = (-P1 * (Math.Log10(P1) / Math.Log10(2))) - ((1 -
P1) * (Math.Log10(1 - P1) / Math.Log10(2)))
    End If
    If (P2 <> 0) And (P2 <> 1) Then
        H2 = (-P2 * (Math.Log10(P2) / Math.Log10(2))) - ((1 -
P2) * (Math.Log10(1 - P2) / Math.Log10(2)))
    End If
    AE(j) = (((cont1 + cont2) / TCasos) * H1) + (((cont3 +
cont4) / TCasos) * H2)

    'MsgBox (H1)
    'MsgBox (H2)
    cont1 = 0
    cont2 = 0
    cont3 = 0
    cont4 = 0
    P1 = 0
    P2 = 0
    H1 = 0
    H2 = 0

    TextBox1.Text = TextBox1.Text & AE(j) & " - " & "(" & j + 1
& ")"

    If AE(j) < max Then
        max = AE(j)
        Posicion = j
    End If
Next j
```

```
End If
Next j
MsgBox(max)
MsgBox(Posicion)
End Sub

Private Sub cmdEPromN_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdEPromN.Click
    Dim i, j, TCasos, cont1, cont2, cont3, cont4, Positivos,
Negativos, Posicion As Integer
    Dim H1, H2, P1, P2, AE(21), max As Decimal
    TextBox1.Text = ""
    TCasos = dgrN.Rows.Count - 3
    For i = 0 To TCasos
        If dgrN.Rows(i).Cells(22).Value = 1 Then
            Positivos = Positivos + 1
        Else
            Negativos = Negativos + 1
        End If
    Next
    max = 10
    For j = 0 To 21
        For i = 0 To TCasos
            If dgrN.Rows(i).Cells(j).Value = 0 Then
                If dgrN.Rows(i).Cells(22).Value = 1 Then
                    cont1 = cont1 + 1
                Else
                    cont2 = cont2 + 1
                End If
            Else
                If dgrN.Rows(i).Cells(22).Value = 1 Then
                    cont3 = cont3 + 1
                Else
                    cont4 = cont4 + 1
                End If
            End If
        Next i
        If (cont1 + cont2) > 0 Then
            P1 = cont1 / (cont1 + cont2)
        End If
        If (cont3 + cont4) > 0 Then
            P2 = cont3 / (cont3 + cont4)
        End If

        'MsgBox(P1)
        'MsgBox(P2)
        If (P1 <> 0) And (P1 <> 1) Then
            H1 = (-P1 * (Math.Log10(P1) / Math.Log10(2))) - ((1 -
P1) * (Math.Log10(1 - P1) / Math.Log10(2)))
        End If
        If (P2 <> 0) And (P2 <> 1) Then
            H2 = (-P2 * (Math.Log10(P2) / Math.Log10(2))) - ((1 -
P2) * (Math.Log10(1 - P2) / Math.Log10(2)))
        End If
        AE(j) = (((cont1 + cont2) / TCasos) * H1) + (((cont3 +
cont4) / TCasos) * H2)
```

```
'MsgBox (H1)
'MsgBox (H2)
cont1 = 0
cont2 = 0
cont3 = 0
cont4 = 0
P1 = 0
P2 = 0
H1 = 0
H2 = 0

TextBox1.Text = TextBox1.Text & AE(j) & " - " & "(" & j + 1
& ")" "
If AE(j) < max Then
    max = AE(j)
    Posicion = j
End If
Next j
MsgBox (max)
MsgBox (Posicion)
End Sub
```

b.3.4 Evaluación de Nuevo caso

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Dim i As Integer
    Dim band As Boolean
    dgrCaso.Rows.Add()
    dgrCaso.Rows.Add()
    band = False
    For i = 0 To 10
        If dgrCaso.Rows(0).Cells(i).Value >= 60 Then
            dgrCaso.Rows(1).Cells(i).Value = 1
        Else
            dgrCaso.Rows(1).Cells(i).Value = 0
        End If
    Next i
    If (dgrCaso.Rows(1).Cells(0).Value = 0 And
dgrCaso.Rows(1).Cells(1).Value = 0 And dgrCaso.Rows(1).Cells(2).Value =
0) Then
        dgrCaso.Rows(1).Cells(11).Value = 0
        dgrCaso.Rows(0).Cells(11).Value = "<60"
        band = True
    End If
    If (dgrCaso.Rows(1).Cells(0).Value = 0 And
dgrCaso.Rows(1).Cells(1).Value = 0 And dgrCaso.Rows(1).Cells(2).Value =
1 And dgrCaso.Rows(1).Cells(3).Value = 0 And
dgrCaso.Rows(1).Cells(4).Value = 1) Then
        dgrCaso.Rows(1).Cells(11).Value = 0
        dgrCaso.Rows(0).Cells(11).Value = "<60"
        band = True
    End If
```

```
If (dgrCaso.Rows(1).Cells(0).Value = 0 And
dgrCaso.Rows(1).Cells(1).Value = 1 And dgrCaso.Rows(1).Cells(3).Value =
1) Then
    dgrCaso.Rows(1).Cells(11).Value = 0
    dgrCaso.Rows(0).Cells(11).Value = "<60"
    band = True

End If

If (dgrCaso.Rows(1).Cells(0).Value = 1 And
dgrCaso.Rows(1).Cells(5).Value = 0) Then
    dgrCaso.Rows(1).Cells(11).Value = 0
    dgrCaso.Rows(0).Cells(11).Value = "<60"
    band = True

End If

If (dgrCaso.Rows(1).Cells(0).Value = 1 And
dgrCaso.Rows(1).Cells(5).Value = 1 And dgrCaso.Rows(1).Cells(6).Value =
0 And dgrCaso.Rows(1).Cells(7).Value = 1 And
dgrCaso.Rows(1).Cells(8).Value = 0 And dgrCaso.Rows(1).Cells(9).Value =
1 And dgrCaso.Rows(1).Cells(10).Value = 1) Then
    dgrCaso.Rows(1).Cells(11).Value = 0
    dgrCaso.Rows(0).Cells(11).Value = "<60"
    band = True

End If

If (dgrCaso.Rows(1).Cells(0).Value = 1 And
dgrCaso.Rows(1).Cells(5).Value = 1 And dgrCaso.Rows(1).Cells(6).Value =
0 And dgrCaso.Rows(1).Cells(7).Value = 1 And
dgrCaso.Rows(1).Cells(8).Value = 1 And dgrCaso.Rows(1).Cells(9).Value =
0) Then
    dgrCaso.Rows(1).Cells(11).Value = 0
    dgrCaso.Rows(0).Cells(11).Value = "<60"
    band = True

End If

If (dgrCaso.Rows(1).Cells(0).Value = 1 And
dgrCaso.Rows(1).Cells(4).Value = 1 And dgrCaso.Rows(1).Cells(5).Value =
1 And dgrCaso.Rows(1).Cells(6).Value = 1 And
dgrCaso.Rows(1).Cells(7).Value = 1 And dgrCaso.Rows(1).Cells(8).Value =
0 And dgrCaso.Rows(1).Cells(10).Value = 0) Then
    dgrCaso.Rows(1).Cells(11).Value = 0
    dgrCaso.Rows(0).Cells(11).Value = "<60"
    band = True

End If
If band = False Then
    dgrCaso.Rows(1).Cells(11).Value = 1
    dgrCaso.Rows(0).Cells(11).Value = ">=60"
End If
End Sub
```

b.4 Vecinos Más Cercano

b.4.1 Selección de Datos

```
Private Sub TSVMC_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TSVMC.Click
    ChildForm3.Show()
    ChildForm1.Hide()
    ChildForm2.Hide()
    ChildForm4.Hide()
    ChildForm5.Hide()
    Dim i, total, a As Integer
    Dim IIC, LP, GC, PG As String
    total = Me.ChildForm4.dgrOrganizacion.Rows.Count - 1
    a = 0
    Me.ChildForm3.dgEstadistica.Rows.Add()

    For i = 0 To 291 'total - 2
        IIC = Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(2).Value
        LP = Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(7).Value
        GC = Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(12).Value
        PG = Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(22).Value
        'If (IIC = "" Or LP = "" Or GC = "" Or PG = "") Then
        'Else
        If (PG = "") Then
        Else
            Me.ChildForm3.dgEstadistica.Rows(a).Cells(0).Value =
IIC
            Me.ChildForm3.dgEstadistica.Rows(a).Cells(1).Value = LP
            Me.ChildForm3.dgEstadistica.Rows(a).Cells(2).Value = GC
            Me.ChildForm3.dgEstadistica.Rows(a).Cells(3).Value = PG
            a = a + 1
            Me.ChildForm3.dgEstadistica.Rows.Add()
        End If
    Next i
    Me.ChildForm3.TotalCasos.Text = a
End Sub
```

b.4.2 Transformación de Datos

```
Private Sub Transformación_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Transformación.Click
    Transformación.Enabled = False
    valor1 = "1000" '91-100
    valor2 = "0100" '80-90
    valor3 = "0010" '70-79
    valor4 = "0001" '60-69
    valor5 = "0000" '0-59
    Nulo = "1111"
    Dim i, j, TCasos As Integer

    TCasos = dgEstadistica.Rows.Count - 3
    dgEstadisticaT.Rows.Add()
    MsgBox(TCasos)
    For i = 0 To TCasos
        For j = 0 To 3
            If dgEstadistica.Rows(i).Cells(j).Value >= 91 Then
```



```
dgEstadisticaT.Rows(i).Cells(j).Value = valor1
ElseIf dgEstadistica.Rows(i).Cells(j).Value >= 80 And
dgEstadistica.Rows(i).Cells(j).Value <= 90 Then
dgEstadisticaT.Rows(i).Cells(j).Value = valor2
ElseIf dgEstadistica.Rows(i).Cells(j).Value >= 70 And
dgEstadistica.Rows(i).Cells(j).Value <= 79 Then
dgEstadisticaT.Rows(i).Cells(j).Value = valor3
ElseIf dgEstadistica.Rows(i).Cells(j).Value >= 60 And
dgEstadistica.Rows(i).Cells(j).Value <= 69 Then
dgEstadisticaT.Rows(i).Cells(j).Value = valor4
ElseIf dgEstadistica.Rows(i).Cells(j).Value = "" Then
dgEstadisticaT.Rows(i).Cells(j).Value = Nulo
ElseIf dgEstadistica.Rows(i).Cells(j).Value <= 59 Then
dgEstadisticaT.Rows(i).Cells(j).Value = valor5

End If
Next j
dgEstadisticaT.Rows.Add()
Next i

End Sub
```

b.4.3 Evaluación de Nuevo Caso

```
Private Sub cmdEvaluarA_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdEvaluarA.Click
Dim i, j, a, C1, C2, C3, C4, C5, C6, TCasos As Integer
valor1 = "1000" '91-100
valor2 = "0100" '80-90
valor3 = "0010" '70-79
valor4 = "0001" '60-69
valor5 = "0000" '0-59
dgEstadisticaV.Rows.Clear()
For j = 0 To 2
If dgEstadisticaE.Rows(0).Cells(j).Value >= 91 Then
dgEstadisticaC.Rows(0).Cells(j).Value = valor1
ElseIf dgEstadisticaE.Rows(0).Cells(j).Value >= 80 And
dgEstadisticaE.Rows(0).Cells(j).Value <= 90 Then
dgEstadisticaC.Rows(0).Cells(j).Value = valor2
ElseIf dgEstadisticaE.Rows(0).Cells(j).Value >= 70 And
dgEstadisticaE.Rows(0).Cells(j).Value <= 79 Then
dgEstadisticaC.Rows(0).Cells(j).Value = valor3
ElseIf dgEstadisticaE.Rows(0).Cells(j).Value >= 60 And
dgEstadisticaE.Rows(0).Cells(j).Value <= 69 Then
dgEstadisticaC.Rows(0).Cells(j).Value = valor4
ElseIf dgEstadisticaE.Rows(0).Cells(j).Value = "" Then
dgEstadisticaC.Rows(0).Cells(j).Value = Nulo
ElseIf dgEstadisticaE.Rows(0).Cells(j).Value <= 59 Then
dgEstadisticaC.Rows(0).Cells(j).Value = valor5
End If
Next j
TCasos = dgEstadisticaT.Rows.Count - 2
dgEstadisticaV.Rows.Add()
a = 0
For i = 0 To TCasos
If dgEstadisticaT.Rows(i).Cells(0).Value =
dgEstadisticaC.Rows(0).Cells(0).Value And
```

```
dgEstadisticaT.Rows(i).Cells(1).Value =  
dgEstadisticaC.Rows(0).Cells(1).Value And  
dgEstadisticaT.Rows(i).Cells(2).Value =  
dgEstadisticaC.Rows(0).Cells(2).Value Then  
    dgEstadisticaV.Rows(a).Cells(0).Value =  
dgEstadisticaT.Rows(i).Cells(0).Value  
    dgEstadisticaV.Rows(a).Cells(1).Value =  
dgEstadisticaT.Rows(i).Cells(1).Value  
    dgEstadisticaV.Rows(a).Cells(2).Value =  
dgEstadisticaT.Rows(i).Cells(2).Value  
    dgEstadisticaV.Rows(a).Cells(3).Value =  
dgEstadisticaT.Rows(i).Cells(3).Value  
    dgEstadisticaV.Rows(a).Cells(4).Value = 3  
    a = a + 1  
    dgEstadisticaV.Rows.Add()  
End If  
Next i  
  
For i = 0 To TCasos  
    If dgEstadisticaT.Rows(i).Cells(0).Value =  
dgEstadisticaC.Rows(0).Cells(0).Value And  
dgEstadisticaT.Rows(i).Cells(1).Value =  
dgEstadisticaC.Rows(0).Cells(1).Value And  
dgEstadisticaT.Rows(i).Cells(2).Value <>  
dgEstadisticaC.Rows(0).Cells(2).Value Then  
        dgEstadisticaV.Rows(a).Cells(0).Value =  
dgEstadisticaT.Rows(i).Cells(0).Value  
        dgEstadisticaV.Rows(a).Cells(1).Value =  
dgEstadisticaT.Rows(i).Cells(1).Value  
        dgEstadisticaV.Rows(a).Cells(2).Value =  
dgEstadisticaT.Rows(i).Cells(2).Value  
        dgEstadisticaV.Rows(a).Cells(3).Value =  
dgEstadisticaT.Rows(i).Cells(3).Value  
        dgEstadisticaV.Rows(a).Cells(4).Value = 2  
        a = a + 1  
        dgEstadisticaV.Rows.Add()  
    End If  
Next i  
  
For i = 0 To TCasos  
    If dgEstadisticaT.Rows(i).Cells(0).Value =  
dgEstadisticaC.Rows(0).Cells(0).Value And  
dgEstadisticaT.Rows(i).Cells(1).Value <>  
dgEstadisticaC.Rows(0).Cells(1).Value And  
dgEstadisticaT.Rows(i).Cells(2).Value =  
dgEstadisticaC.Rows(0).Cells(2).Value Then  
        dgEstadisticaV.Rows(a).Cells(0).Value =  
dgEstadisticaT.Rows(i).Cells(0).Value  
        dgEstadisticaV.Rows(a).Cells(1).Value =  
dgEstadisticaT.Rows(i).Cells(1).Value  
        dgEstadisticaV.Rows(a).Cells(2).Value =  
dgEstadisticaT.Rows(i).Cells(2).Value  
        dgEstadisticaV.Rows(a).Cells(3).Value =  
dgEstadisticaT.Rows(i).Cells(3).Value  
        dgEstadisticaV.Rows(a).Cells(4).Value = 2  
        a = a + 1  
        dgEstadisticaV.Rows.Add()
```

```
End If
Next i

For i = 0 To TCasos
    If dgEstadisticaT.Rows(i).Cells(0).Value <>
dgEstadisticaC.Rows(0).Cells(0).Value And
dgEstadisticaT.Rows(i).Cells(1).Value =
dgEstadisticaC.Rows(0).Cells(1).Value And
dgEstadisticaT.Rows(i).Cells(2).Value =
dgEstadisticaC.Rows(0).Cells(2).Value Then
        dgEstadisticaV.Rows(a).Cells(0).Value =
dgEstadisticaT.Rows(i).Cells(0).Value
        dgEstadisticaV.Rows(a).Cells(1).Value =
dgEstadisticaT.Rows(i).Cells(1).Value
        dgEstadisticaV.Rows(a).Cells(2).Value =
dgEstadisticaT.Rows(i).Cells(2).Value
        dgEstadisticaV.Rows(a).Cells(3).Value =
dgEstadisticaT.Rows(i).Cells(3).Value
        dgEstadisticaV.Rows(a).Cells(4).Value = 2
        a = a + 1
        dgEstadisticaV.Rows.Add()
    End If
Next i

Dim var As String
If CVecinos.Text = 1 Or CVecinos.Text = 2 Then
    var = dgEstadisticaV.Rows(0).Cells(3).Value
    dgEstadisticaC.Rows(0).Cells(3).Value = var
Else
    For i = 0 To CVecinos.Text - 1
        If dgEstadisticaV.Rows(i).Cells(3).Value = "1000" Then
            C1 = C1 + 1
        End If
        If dgEstadisticaV.Rows(i).Cells(3).Value = "0100" Then
            C2 = C2 + 1
        End If
        If dgEstadisticaV.Rows(i).Cells(3).Value = "0010" Then
            C3 = C3 + 1
        End If
        If dgEstadisticaV.Rows(i).Cells(3).Value = "0001" Then
            C4 = C4 + 1
        End If
        If dgEstadisticaV.Rows(i).Cells(3).Value = "0000" Then
            C5 = C5 + 1
        End If
        If dgEstadisticaV.Rows(i).Cells(3).Value = "1111" Then
            C6 = C6 + 1
        End If
    Next i
End If
If C1 = 0 And C2 = 0 And C3 = 0 And C4 = 0 And C5 = 0 And C6 =
0 Then
Else
    Dim maximo1, maximo2, maximo3, maximo4, maximo5 As Integer
    maximo1 = Math.Max(C1, C2)
    maximo2 = Math.Max(C3, C4)
    maximo3 = Math.Max(C5, C6)
```

```
maximo4 = Math.Max(maximo1, maximo2)
maximo5 = Math.Max(maximo4, maximo3)

If maximo5 = C1 Then
    dgEstadisticaC.Rows(0).Cells(3).Value = "1000"
ElseIf maximo5 = C2 Then
    dgEstadisticaC.Rows(0).Cells(3).Value = "0100"
ElseIf maximo5 = C3 Then
    dgEstadisticaC.Rows(0).Cells(3).Value = "0010"
ElseIf maximo5 = C4 Then
    dgEstadisticaC.Rows(0).Cells(3).Value = "0001"
ElseIf maximo5 = C5 Then
    dgEstadisticaC.Rows(0).Cells(3).Value = "0000"
ElseIf maximo5 = C6 Then
    dgEstadisticaC.Rows(0).Cells(3).Value = "1111"
End If
End If

If dgEstadisticaC.Rows(0).Cells(3).Value = "1000" Then
    dgEstadisticaE.Rows(0).Cells(3).Value = "91-100"
ElseIf dgEstadisticaC.Rows(0).Cells(3).Value = "0100" Then
    dgEstadisticaE.Rows(0).Cells(3).Value = "80-90"
ElseIf dgEstadisticaC.Rows(0).Cells(3).Value = "0010" Then
    dgEstadisticaE.Rows(0).Cells(3).Value = "70-79"
ElseIf dgEstadisticaC.Rows(0).Cells(3).Value = "0001" Then
    dgEstadisticaE.Rows(0).Cells(3).Value = "60-69"
ElseIf dgEstadisticaC.Rows(0).Cells(3).Value = "0000" Then
    dgEstadisticaE.Rows(0).Cells(3).Value = "0-59"
ElseIf dgEstadisticaC.Rows(0).Cells(3).Value = "1111" Then
    dgEstadisticaE.Rows(0).Cells(3).Value = ""
End If
End Sub
```

b.4.4 Evaluación de Nuevo Caso con distancia Manhattan

```
Private Sub Evaluar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Evaluar.Click
    Dim i, prom, sumador As Integer

    'dgEstadisticaE1.Rows(0).Cells(3).Value =
DGEstadistica1.Rows(0).Cells(3).Value
    For i = 0 To CVecinos1.Text - 1
        sumador = sumador + DGEstadistica1.Rows(i).Cells(3).Value
    Next
    prom = sumador / CVecinos1.Text
    dgEstadisticaE1.Rows(0).Cells(3).Value = prom
End Sub
```

B.4.5 Cálculo de Distancia

```
Private Sub Distancia_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Distancia.Click
    Dim TCasos, i, j As Integer
    Dim N1, N2, N3, T1, T2, T3, Total As Integer
    N1 = dgEstadisticaE1.Rows(0).Cells(0).Value
    N2 = dgEstadisticaE1.Rows(0).Cells(1).Value
    N3 = dgEstadisticaE1.Rows(0).Cells(2).Value
```

```
DGEstadistical.Rows.Clear()  
TCasos = dgEstadistica.Rows.Count - 3  
For i = 0 To TCasos  
    DGEstadistical.Rows.Add()  
    T1 = dgEstadistica.Rows(i).Cells(0).Value - N1  
    If T1 < 0 Then  
        T1 = T1 * -1  
    End If  
    T2 = dgEstadistica.Rows(i).Cells(1).Value - N2  
    If T2 < 0 Then  
        T2 = T2 * -1  
    End If  
    T3 = dgEstadistica.Rows(i).Cells(2).Value - N3  
    If T3 < 0 Then  
        T3 = T3 * -1  
    End If  
  
    Total = (T1 + T2 + T3)  
  
    DGEstadistical.Rows(i).Cells(0).Value = T1  
    DGEstadistical.Rows(i).Cells(1).Value = T2  
    DGEstadistical.Rows(i).Cells(2).Value = T3  
    DGEstadistical.Rows(i).Cells(3).Value =  
dgEstadistica.Rows(i).Cells(3).Value  
    DGEstadistical.Rows(i).Cells(4).Value = Total  
  
Next i  
End Sub
```

b.4.5 Ordenar Distancias

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Ordenar.Click  
    DGEstadistical.Sort(DGEstadistical.Columns(4),  
System.ComponentModel.ListSortDirection.Ascending)  
    Evaluar.Enabled = True  
End Sub
```

b.5 Validación Cruzada

b.5.1 Selección de Datos

```
Private Sub TSVC_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TSVC.Click
    ChildForm3.Hide()
    ChildForm1.Hide()
    ChildForm2.Hide()
    ChildForm5.Show()
    ChildForm4.Hide()

    Dim i, total, a As Integer
    Dim MI, MII, MIII, ES As String
    total = Me.ChildForm4.dgrOrganizacion.Rows.Count - 1
    a = 0
    ChildForm5.dgr7.Rows.Add()
    For i = 0 To total - 2

        If
IsDBNull(Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(5).Value) Then
            MI = ""
        Else
            MI =
Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(5).Value
        End If
        If
IsDBNull(Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(10).Value) Then
            MII = ""
        Else
            MII =
Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(10).Value
        End If
        MIII =
Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(14).Value
        If
IsDBNull(Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(23).Value) Then
            ES = ""
        Else
            ES =
Me.ChildForm4.dgrOrganizacion.Rows(i).Cells(23).Value
        End If

        If (MI = "" Or MII = "" Or MIII = "" Or ES = "") Then
        Else
            ChildForm5.dgr7.Rows(a).Cells(0).Value = MI
            ChildForm5.dgr7.Rows(a).Cells(1).Value = MII
            ChildForm5.dgr7.Rows(a).Cells(2).Value = MIII
            ChildForm5.dgr7.Rows(a).Cells(3).Value = ES
            a = a + 1
            ChildForm5.dgr7.Rows.Add()
        End If
    Next i
End Sub
```

b.5.2 Transformación de Datos

```
Private Sub cmdTransformacion Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdTransformacion.Click
    Dim i, j, TCasos As Integer
    cmdTransformacion.Enabled = False
    TCasos = dgr7.Rows.Count - 3
    dgrTransformacion.Rows.Add()
    For i = 0 To TCasos
        For j = 0 To 3
            If dgr7.Rows(i).Cells(j).Value >= 60 Then
                dgrTransformacion.Rows(i).Cells(j).Value = 1
            Else
                dgrTransformacion.Rows(i).Cells(j).Value = 0
            End If
        Next j
        dgrTransformacion.Rows.Add()
    Next i
End Sub
```

b.5.3 Generación de Hipótesis

```
Private Sub cmdGHip_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdGHip.Click
    Dim i, j, TCasos, cont1, cont2, cont3, cont4 As Integer
    cmdGHip.Enabled = False
    TCasos = dgrTransformacion.Rows.Count - 3
    ReDim cantidad(3, 4)
    ReDim cantidad2(3, 4)
    ReDim cantidad3(3, 4)

    For i = 0 To 29
        If dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
            positivos = positivos + 1
        Else
            negativos = negativos + 1
        End If
    Next
    For j = 0 To 2
        For i = 0 To 29
            If dgrTransformacion.Rows(i).Cells(j).Value = 1 And
dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
                cont1 = cont1 + 1
            ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 0 And
dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
                cont2 = cont2 + 1
            ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 1 And
dgrTransformacion.Rows(i).Cells(3).Value = 0 Then
                cont3 = cont3 + 1
            ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 0 And
dgrTransformacion.Rows(i).Cells(3).Value = 0 Then
                cont4 = cont4 + 1
            End If
        Next i
        cantidad(j, 0) = cont1
        cantidad(j, 1) = cont2
        cantidad(j, 2) = cont3
```



```
        cantidad(j, 3) = cont4
        cont1 = 0
        cont2 = 0
        cont3 = 0
        cont4 = 0
    Next j

    For i = 0 To 3
        Hipotesis.Rows.Add()
    Next
    Hipotesis.Rows(3).DefaultCellStyle.BackColor =
Color.LightSteelBlue

    For i = 0 To 2
        Hipotesis.Rows(i).Cells(0).Value = i + 1
        Hipotesis.Rows(i).Cells(1).Value = cantidad(i, 0) + 1 & "/"
& positivos + 2
        Hipotesis.Rows(i).Cells(2).Value = cantidad(i, 1) + 1 & "/"
& positivos + 2
        Hipotesis.Rows(i).Cells(3).Value = cantidad(i, 2) + 1 & "/"
& negativos + 2
        Hipotesis.Rows(i).Cells(4).Value = cantidad(i, 3) + 1 & "/"
& negativos + 2
    Next
    !*****
    For i = 30 To 59
        If dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
            positivos2 = positivos2 + 1
        Else
            negativos2 = negativos2 + 1
        End If
    Next
    For j = 0 To 2
        For i = 30 To 59
            If dgrTransformacion.Rows(i).Cells(j).Value = 1 And
dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
                cont1 = cont1 + 1
            ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 0 And
dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
                cont2 = cont2 + 1
            ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 1 And
dgrTransformacion.Rows(i).Cells(3).Value = 0 Then
                cont3 = cont3 + 1
            ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 0 And
dgrTransformacion.Rows(i).Cells(3).Value = 0 Then
                cont4 = cont4 + 1
            End If
        Next i
        cantidad2(j, 0) = cont1
        cantidad2(j, 1) = cont2
        cantidad2(j, 2) = cont3
        cantidad2(j, 3) = cont4
        cont1 = 0
        cont2 = 0
        cont3 = 0
        cont4 = 0
    Next j
```

```
For i = 0 To 3
    Hipotesis.Rows.Add()
Next

Hipotesis.Rows(7).DefaultCellStyle.BackColor =
Color.LightSteelBlue

For i = 0 To 2
    Hipotesis.Rows(i + 4).Cells(0).Value = i + 1
    Hipotesis.Rows(i + 4).Cells(1).Value = cantidad2(i, 0) + 1
    & "/" & positivos2 + 2
    Hipotesis.Rows(i + 4).Cells(2).Value = cantidad2(i, 1) + 1
    & "/" & positivos2 + 2
    Hipotesis.Rows(i + 4).Cells(3).Value = cantidad2(i, 2) + 1
    & "/" & negativos2 + 2
    Hipotesis.Rows(i + 4).Cells(4).Value = cantidad2(i, 3) + 1
    & "/" & negativos2 + 2
Next
!*****
For i = 60 To 89
    If dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
        positivos3 = positivos3 + 1
    Else
        negativos3 = negativos3 + 1
    End If
Next
For j = 0 To 2
    For i = 60 To 89
        If dgrTransformacion.Rows(i).Cells(j).Value = 1 And
dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
            cont1 = cont1 + 1
        ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 0 And
dgrTransformacion.Rows(i).Cells(3).Value = 1 Then
            cont2 = cont2 + 1
        ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 1 And
dgrTransformacion.Rows(i).Cells(3).Value = 0 Then
            cont3 = cont3 + 1
        ElseIf dgrTransformacion.Rows(i).Cells(j).Value = 0 And
dgrTransformacion.Rows(i).Cells(3).Value = 0 Then
            cont4 = cont4 + 1
        End If
    Next i
    cantidad3(j, 0) = cont1
    cantidad3(j, 1) = cont2
    cantidad3(j, 2) = cont3
    cantidad3(j, 3) = cont4
    cont1 = 0
    cont2 = 0
    cont3 = 0
    cont4 = 0
Next j

For i = 0 To 3
    Hipotesis.Rows.Add()
Next
```

```
Hipotesis.Rows(11).DefaultCellStyle.BackColor =  
Color.LightSteelBlue  
  
    For i = 0 To 2  
        Hipotesis.Rows(i + 8).Cells(0).Value = i + 1  
        Hipotesis.Rows(i + 8).Cells(1).Value = cantidad3(i, 0) + 1  
& "/" & positivos3 + 2  
        Hipotesis.Rows(i + 8).Cells(2).Value = cantidad3(i, 1) + 1  
& "/" & positivos3 + 2  
        Hipotesis.Rows(i + 8).Cells(3).Value = cantidad3(i, 2) + 1  
& "/" & negativos3 + 2  
        Hipotesis.Rows(i + 8).Cells(4).Value = cantidad3(i, 3) + 1  
& "/" & negativos3 + 2  
    Next  
  
End Sub
```

b.5.4 Selección de Nuestro Conjunto de Datos

```
Private Sub cmdCPrueba_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdCPrueba.Click  
    dgrVal.Rows.Clear()  
    Dim i, j, TCasos, a As Integer  
    TCasos = dgrTransformacion.Rows.Count - 3  
    dgrVal.Rows.Add()  
    For i = 90 To TCasos  
        For j = 0 To 3  
            dgrVal.Rows(a).Cells(j).Value =  
dgr7.Rows(i).Cells(j).Value  
        Next  
        a = a + 1  
        dgrVal.Rows.Add()  
    Next  
End Sub
```

b.5.5 Evaluación de Casos

```
Private Sub cmdECasos_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdECasos.Click  
    Dim i, j, Error1, Error2, Error3 As Integer  
    Dim s1, s2, s12, s13, s22, s23 As Decimal  
    Dim casos1, casos2, casos3 As String  
    dgCaso.Rows.Clear()  
    s1 = 1 : s2 = 1 : s12 = 1 : s13 = 1 : s22 = 1 : s23 = 1  
  
    For i = 0 To 19  
        dgCaso.Rows.Add()  
    Next  
  
    For j = 0 To dgrVal.Rows.Count - 3  
        For i = 0 To 3  
            If dgrVal.Rows(j).Cells(i).Value >= 60 Then  
                dgCaso.Rows(j).Cells(i).Value = 1  
            Else  
                dgCaso.Rows(j).Cells(i).Value = 0  
            End If  
        Next  
    Next
```

```
Next
For i = 0 To 2
    If dgCaso.Rows(j).Cells(i).Value = 1 Then
        s1 = s1 * ((cantidad(i, 0) + 1) / (positivos + 2))
        s12 = s12 * ((cantidad2(i, 0) + 1) / (positivos2 +
2))
        s13 = s13 * ((cantidad3(i, 0) + 1) / (positivos3 +
2))

    Else
        s1 = s1 * ((cantidad(i, 1) + 1) / (positivos + 2))
        s12 = s12 * ((cantidad2(i, 1) + 1) / (positivos2 +
2))
        s13 = s13 * ((cantidad3(i, 1) + 1) / (positivos3 +
2))

    End If
Next i

For i = 0 To 2
    If dgCaso.Rows(j).Cells(i).Value = 1 Then
        s2 = s2 * ((cantidad(i, 2) + 1) / (negativos + 2))
        s22 = s22 * ((cantidad2(i, 2) + 1) / (negativos2 +
2))
        s23 = s23 * ((cantidad3(i, 2) + 1) / (negativos3 +
2))

    Else
        s2 = s2 * ((cantidad(i, 3) + 1) / (negativos + 2))
        s22 = s22 * ((cantidad2(i, 3) + 1) / (negativos2 +
2))
        s23 = s23 * ((cantidad3(i, 3) + 1) / (negativos3 +
2))

    End If
Next i
If s1 > s2 Then
    dgCaso.Rows(j).Cells(4).Value = 1
Else
    dgCaso.Rows(j).Cells(4).Value = 0
End If
If s12 > s22 Then
    dgCaso.Rows(j).Cells(5).Value = 1
Else
    dgCaso.Rows(j).Cells(5).Value = 0
End If
If s13 > s23 Then
    dgCaso.Rows(j).Cells(6).Value = 1
Else
    dgCaso.Rows(j).Cells(6).Value = 0
End If
Next j
casos1 = ""
casos2 = ""
casos3 = ""
For i = 0 To 19
```

```
        If dgCaso.Rows(i).Cells(3).Value <>
dgCaso.Rows(i).Cells(4).Value Then
            Error1 = Error1 + 1
            casos1 = casos1 & " " & i + 1 & " ,"
        End If
        If dgCaso.Rows(i).Cells(3).Value <>
dgCaso.Rows(i).Cells(5).Value Then
            Error2 = Error2 + 1
            casos2 = casos2 & " " & i + 1 & " ,"
        End If
        If dgCaso.Rows(i).Cells(3).Value <>
dgCaso.Rows(i).Cells(6).Value Then
            Error3 = Error3 + 1
            casos3 = casos3 & " " & i + 1 & " ,"
        End If
    Next

    Label3.Text = "La hipótesis 1 falla un total de: " & Error1 &
"/20 " & "Casos. N° " & casos1
    Label4.Text = "La hipótesis 2 falla un total de: " & Error2 &
"/20 " & "Casos. N° " & casos2
    Label5.Text = "La hipótesis 3 falla un total de: " & Error3 &
"/20 " & "Casos. N° " & casos3

End Sub
```